

AI FOR MUSICIANS

A Complete Course in AI Literacy, Creative Tools,
Web Development & Deployment for Music Students

One Hour a Day. Eleven Weeks. Zero Prior Experience.

A Complete, Self-Paced Course for Creative Minds

by Greg Spero

Updated April 2026

Table of Contents

Part I — Welcome & Orientation	6
1. Who This Handbook Is For	6
2. How This Course Works	7
3. What You Will Be Able to Do in Eleven Weeks	8
4. Equipment & Setup	9
 Part II — The Course	11
Week 1: AI Fundamentals & Prompt Mastery	11
Day 1: What AI Actually Is	11
Day 2: Meeting Claude — Your AI Partner	14
Day 3: The Science of Prompting	16
Day 4: Advanced Prompting Techniques	20
Day 5: Comparing AI Models	23
Day 6: AI Ethics, Bias & Limitations	26
Day 7: Review, Practice & Reflection	29
 Week 2: AI for Writing, Research & Communication	31
Day 1: Writing Program Notes, Bios & Artist Statements	31
Day 2: Research with AI — Music History & Theory	34
Day 3: Academic Writing — Papers, Theses & Applications	35
Day 4: Professional Communication — Emails, Proposals & Contracts	31
Day 5: Social Media & Marketing Content	39
Day 6: Multi-Step Workflows & Chain-of-Thought	40
Day 7: Review & Portfolio Assembly	42
 Week 3: AI Audio & Music Creation	43
Day 1: AI Stem Separation — Deconstructing Music	43
Day 2: AI Music Generation — Capabilities, Tools & Ethics	45
Day 3: AI Mastering & Production Enhancement	47
Day 4: AI Transcription, Notation & Lead Sheets	49
Day 5: AI Voice Synthesis & Narration	50

Day 6: AI Audio Editing & Podcast Production	45
Day 7: Building Your AI Music Production Workflow	53
Week 4: AI Visual & Video Creation	54
Day 1: AI Image Generation Fundamentals	54
Day 2: Album Art, Posters & Promotional Graphics	56
Day 3: AI Video Generation for Musicians	57
Day 4: Music Videos & Visual Content with AI	58
Day 5: AI Presentations & Pitch Decks	59
Day 6: Branding & Visual Identity with AI	60
Day 7: Review & Visual Portfolio	42
Week 5: AI Automation, Workflows & Business Tools	62
Day 1: No-Code AI Platforms & Why They Matter	62
Day 2: Connecting AI Tools Together	63
Day 3: AI for Scheduling, Email & Admin	64
Day 4: Building an AI-Powered Electronic Press Kit	64
Day 5: AI for Teaching & Pedagogy	65
Day 6: AI for Live Performance & Shows	67
Day 7: Assembling Your Personal AI Toolkit	67
Week 6: Building Your Website with Claude Code	69
Day 1: Why Every Musician Needs a Website (and Full Control of It)	69
Day 2: The Terminal & Setting Up Your Tools	70
Day 3: Meet Claude Code — Your AI Development Partner	69
Day 4: Building Your Site with Claude Code	64
Day 5: Deploying to Vercel or Cloudflare Pages	76
Day 6: Custom Domains, SEO & Email Lists	78
Day 7: Launch Day	80
Week 7: AI for Your Music Career & Business	81
Day 1: AI for Grant Writing & Funding Applications	35
Day 2: Financial Planning & Budgeting with AI	82

Day 3: Contract Review & Negotiation Support	83
Day 4: Networking & Relationship Management	83
Day 5: AI for Tour Planning & Logistics	65
Day 6: Staying Current — The AI Landscape Evolves	84
Day 7: Your AI-Powered Career Plan	84
Week 8: How the Web Works & Your Development Environment	85
Day 1: How the Internet Actually Works	85
Day 2: The Terminal — Your New Command Center	70
Day 3: Setting Up Your Development Environment	85
Day 4: Claude Code — Your AI Development Partner	69
Day 5: Your First Web Page with Claude Code	69
Day 6: Git & Version Control	91
Day 7: Review & Confidence Building	42
Week 9: Building Real Applications with Claude Code	35
Day 1: How Web Applications Work	35
Day 2: Your First Backend — Node.js & Express	94
Day 3: Databases — Storing Real Data	95
Day 4: Connecting Frontend to Backend	97
Day 5: API Keys, Environment Variables & Security	98
Day 6: Building a Complete Application	100
Day 7: Testing, Debugging & Code Review	100
Week 10: Deployment & Going Live	102
Day 1: Understanding Servers & Hosting	102
Day 2: Deploying a Static Site	103
Day 3: Deploying a Full-Stack Application	103
Day 4: Databases in Production	104
Day 5: Custom Domains, HTTPS & DNS	105
Day 6: Monitoring, Logs & Maintenance	106
Day 7: Your Complete Development Workflow	107

Week 11: Capstone Projects	108
Day 1: Choosing Your Capstone	108
Day 2: Planning & Architecture	
Days 3–5: Building Your Project	
Day 6: Polish, Test & Present	
Day 7: Reflection & What Comes Next	109
 Part III — Appendices	 111
Appendix A — The Complete AI Prompt Library for Musicians	111
Appendix B — AI Tool Directory & Comparison Charts	116
Appendix C — Troubleshooting & FAQ	120
Appendix D — Glossary	122
Appendix E — Recommended Resources & Further Learning	125
Appendix F — Weekly Checklists & Progress Tracker	127

Part I — Welcome & Orientation

1. Who This Handbook Is For

You are a music student. Your days are built around practice rooms, theory classes, ensemble rehearsals, lessons, and recitals. You chose music because it moves you — because there is nothing else you would rather spend your life doing. You did not sign up to become a technologist, and this handbook does not ask you to become one.

But the world you are graduating into has changed. In the span of just a few years, artificial intelligence has gone from a science fiction concept to an everyday tool that is reshaping how music is created, distributed, marketed, taught, and experienced. The musicians who will thrive in this landscape are not necessarily the ones who can code — they are the ones who understand how to use AI as a creative collaborator, a business partner, and a force multiplier for their artistic vision.

This handbook is for you if any of the following are true:

- You have never used an AI tool, or you have tried ChatGPT once or twice but do not really understand what you are doing
- You are curious about AI but feel overwhelmed by the flood of tools, terms, and opinions
- You want to use AI to help with your music career — writing program notes, creating promotional materials, managing your business — but do not know where to start
- You have heard about AI music generation tools and want to understand what they can do, what they cannot do, and what the ethical implications are
- You want a personal website, professional online presence, and marketing strategy but have no technical background
- You are a music educator who wants to integrate AI literacy into your teaching
- You want to understand AI well enough to have informed opinions about how it affects your art form

This handbook assumes zero prior technical knowledge. Not "some familiarity with computers" — genuinely zero. Every concept is explained from the ground up, every step is spelled out, and every exercise is designed to be completed in a single sitting. If you can check your email and browse the internet, you have all the technical skill you need to begin.

What this handbook is not: it is not a software engineering bootcamp. The original version of this curriculum spent the majority of its time teaching you to code websites from scratch — HTML, CSS, JavaScript, databases, server deployment. Those are valuable skills, but they take years to master, and in 2026, AI tools can build you a professional website in an afternoon without writing a single line of code. This version focuses on what matters most for working musicians right now: understanding AI deeply enough to use it powerfully across every aspect of your creative and professional life.

2. How This Course Works

The One-Hour-a-Day Framework

Every lesson in this handbook is designed for roughly sixty minutes. You can do more if you are on a roll, but the course is built so that one focused hour per day is enough to make real progress. Here is how a typical session breaks down:

- Minutes 1–5: Review yesterday's lesson and any notes you made
- Minutes 5–15: Read the day's new concepts (the explanation portion)
- Minutes 15–45: Hands-on exercises and guided practice
- Minutes 45–55: Free exploration — try variations, experiment, follow your curiosity
- Minutes 55–60: Journal entry — write down what you learned, what surprised you, what confused you

The journal is not optional. Writing about what you learn is one of the most effective ways to consolidate knowledge. It does not need to be long — three to five sentences per day is plenty. Use a notebook, a notes app, or a Google Doc. The format does not matter; the habit does.

The Eleven-Week Structure

The course is organized into eleven weeks, each with seven daily lessons. That is 77 lessons total. The weeks build on each other, but many also stand alone as coherent units. If you are particularly interested in AI audio tools, you could start with Week 3. If you need a website urgently, you could jump to Week 6. But the recommended path is sequential, because the prompting skills you build in Weeks 1 and 2 make everything that follows more effective.

- Week 1: AI Fundamentals & Prompt Mastery — how AI works, how to talk to it effectively
- Week 2: AI for Writing & Communication — program notes, bios, grants, emails, social media
- Week 3: AI Audio & Music Creation — stem separation, music generation, mastering, transcription
- Week 4: AI Visual & Video Creation — album art, music videos, posters, branding
- Week 5: AI Automation & Business Tools — workflows, EPKs, teaching tools, admin
- Week 6: Building Your Website with Claude Code — terminal, Claude Code, build and deploy your site
- Week 7: AI for Your Music Career — grants, contracts, finances, networking, touring
- Week 8: How the Web Works & Your Development Environment — terminal, tools, Claude Code
- Week 9: Building Real Applications with Claude Code — backend, databases, APIs, security
- Week 10: Deployment & Going Live — servers, hosting, domains, monitoring

- Week 11: Capstone Projects — putting it all together into something real

How to Use AI While Learning About AI

Here is something beautifully recursive about this course: your primary learning tool is the very thing you are learning about. From Day 1, you will use Claude (and other AI tools) to help you learn. If a concept does not make sense, ask Claude to explain it differently. If an exercise feels too hard, ask Claude for hints. If you want to go deeper on a topic, ask Claude to teach you more.

This is not cheating — it is the entire point. Learning to use AI effectively means learning to learn with AI. The students who get the most out of this course are the ones who treat Claude as a patient, tireless tutor who is available 24 hours a day and never judges you for asking a basic question.

3. What You Will Be Able to Do in Eleven Weeks

By the end of this course, you will be able to:

AI Fluency — Carry on sophisticated, multi-turn conversations with AI models. Write prompts that get excellent results on the first try. Understand what AI can and cannot do, and explain this to others.

Creative Writing — Use AI to draft program notes, artist bios, grant applications, press releases, social media content, academic papers, and professional emails — all in your own voice.

Music Production — Use AI tools to separate stems from recordings, generate backing tracks, master your recordings, transcribe music to notation, synthesize voices, and edit audio by editing text.

Visual Creation — Generate album art, concert posters, social media graphics, and promotional materials using AI image generators. Create music videos and visual content with AI video tools.

Web Presence — Build and launch a professional musician website using Claude Code as your AI development partner. Deploy it on Vercel or Cloudflare Pages for free. Implement SEO, set up an email list, and create a content strategy.

Business Operations — Use AI to write grant applications, review contracts, plan tours, manage finances, maintain relationships with venues and collaborators, and handle the administrative side of a music career.

Automation — Connect AI tools together into workflows that save you hours every week on repetitive tasks like email responses, social media scheduling, and content creation.

Application Development — Understand how web applications work — frontend, backend, databases, and APIs. Use Claude Code as your AI development partner to build and deploy real applications on live servers that anyone can visit.

Critical Thinking — Evaluate AI-generated content critically. Understand the ethical implications of AI in music. Make informed decisions about when and how to use AI in your creative work.

Teaching — If you are or plan to be a music educator, integrate AI tools into your pedagogy in ways that enhance rather than replace learning.

4. Equipment & Setup

You need the following to complete this course. Everything is either free or has a free tier sufficient for learning:

Required

- A laptop or desktop computer — Mac or Windows. (Chromebooks have limited support for the development tools used in Weeks 6 and 8-10.) A tablet is not sufficient for several exercises.
- A stable internet connection — AI tools require internet access. Most exercises cannot be done offline.
- A free Claude account — Go to claude.ai and sign up with your email. Claude is made by Anthropic and is the primary AI tool used throughout this course. The free tier gives you enough messages per day for every exercise.
- A web browser — Chrome, Firefox, Safari, or Edge. Chrome is recommended because some tools work best in it.
- A notes app or notebook — For your daily journal. Google Docs, Apple Notes, Notion, or a physical notebook all work.
- Headphones or speakers — Essential for the audio/music creation exercises in Week 3.

Recommended (Free Tier Sufficient)

- A ChatGPT account (chatgpt.com) — Used for comparison exercises in Week 1. Free tier is fine.
- A Google account — For Google Gemini access and various Google-integrated tools.
- A Canva account (canva.com) — Used in Week 4 for graphic design. Free tier is generous.
- A Moises.ai account — Used in Week 3 for stem separation. Free tier allows several separations per month.

Setup Instructions

Before starting Day 1, complete these setup steps. They should take about 15 minutes total.

Step 1: Create your Claude account. Go to claude.ai in your web browser. Click 'Sign Up.' Enter your email address and create a password, or sign up with your Google account.

You will receive a verification email — click the link in it. You are now on the free tier, which gives you a generous number of messages per day. You do not need to pay for a Pro account to complete this course, though if you find yourself hitting message limits frequently, the Pro tier removes those limits.

Step 2: Create your learning journal. Open whatever notes app you prefer (or grab a physical notebook). Create a new document or section called 'AI Handbook Journal.' Add today's date. Write: 'Starting the AI for Musicians course. My goal is [write your goal here].' This journal will be valuable to you — it tracks your growth and captures insights you will want to reference later.

Step 3: Create a bookmarks folder. In your browser, create a new bookmarks folder called 'AI Tools.' Bookmark claude.ai as the first entry. You will add many more throughout the course.

Step 4: Test your audio setup. Put on your headphones or turn on your speakers. Play a song you know well. Confirm you can hear it clearly. You will need working audio for Week 3.

You are ready to begin.

Part II — The Course

Week 1: AI Fundamentals & Prompt Mastery

This week is the foundation for everything that follows. You are going to develop a deep, practical understanding of what artificial intelligence is, how it works, what it can and cannot do, and — most importantly — how to communicate with it effectively. By Sunday, you will be writing prompts that consistently produce excellent results, and you will understand why they work.

Day 1: What AI Actually Is

The Concept

Artificial intelligence, as you encounter it in 2026, is not what science fiction taught you to expect. It is not a sentient being. It does not think, feel, want, or understand in the way that you do. What it does — and this is remarkable enough without any mystification — is predict patterns in language with extraordinary sophistication.

Here is the most useful analogy for a musician: imagine a jazz musician who has listened to every recording ever made, memorized every lick, every harmonic pattern, every rhythmic figure, every stylistic tendency of every player. When this musician sits in on a jam session and hears a ii-V-I progression in B-flat, they can predict what notes would sound good next — not because they understand music theory in an abstract sense, but because they have internalized millions of examples of what musicians play over that progression. They can generate something that sounds musical, creative, even inspired — but they are drawing on pattern recognition, not conscious understanding.

That is, roughly, how a large language model (LLM) like Claude works. Instead of musical recordings, it was trained on an enormous amount of text — books, articles, websites, code, conversations, scientific papers. During training, the model learned the statistical patterns of how words, sentences, and ideas relate to each other. When you type a message to Claude, it generates a response by predicting, word by word, what text would most appropriately follow your input, based on all those patterns.

This is called a transformer model, and the specific technique is called next-token prediction. A 'token' is a chunk of text — roughly a word or part of a word. The model looks at everything you have written, processes it through billions of mathematical parameters (think of these as the model's 'trained intuitions'), and generates the most

likely next token. Then it looks at everything including that new token and generates the next one. And so on, token by token, until the response is complete.

What This Means for You

Understanding this mechanism — pattern prediction, not thinking — changes how you use AI. Here are the practical implications:

AI is not looking things up — When you ask Claude a question, it is not searching a database or the internet. It is generating a response based on patterns it learned during training. This is why it can sometimes produce information that sounds confident but is wrong — a phenomenon called 'hallucination.' The model is not lying; it is generating the most likely-seeming text, and sometimes the most likely-seeming text is not factually accurate.

Context is everything — The quality of AI output depends enormously on the quality of your input. The model responds to patterns in your text. If you write vague, generic prompts, you get vague, generic responses. If you write specific, detailed prompts with clear context, you get specific, useful responses. This is why 'prompt engineering' — the skill of writing effective prompts — is the single most important skill you will learn in this course.

AI has no memory between conversations — Each conversation with Claude starts fresh. It does not remember what you discussed yesterday unless you tell it. Within a single conversation, it remembers everything said so far. But close that conversation and start a new one, and it is a blank slate. This means you need to provide context at the start of each new conversation. Claude Pro users can use 'Projects' to maintain persistent context, but even then, understanding this limitation is important.

AI is a tool, not an oracle — AI does not have opinions, preferences, or expertise in the way a human expert does. When Claude says 'I think the best approach is...' it is generating text that follows the pattern of helpful advice — it is not expressing a considered judgment. This means you should treat AI output as a starting point for your own thinking, not as a final answer. The musician's ear, the artist's judgment, the human's ethical compass — these remain yours.

Key Terms

Artificial Intelligence (AI) — The broad field of creating systems that can perform tasks that typically require human intelligence. In casual use, 'AI' almost always refers to large language models like Claude, ChatGPT, or Gemini.

Large Language Model (LLM) — The specific type of AI you will be working with. A model trained on vast amounts of text that generates responses by predicting the most likely next words.

Training — The process by which an AI model learns patterns from data. Claude was trained on text data, learning statistical relationships between words, concepts, and

ideas. Training happens before you ever interact with the model — it is not learning from your conversations in real time.

Token — A chunk of text, roughly equivalent to a word or part of a word. 'Musical' is one token. 'Ethnomusicological' might be two or three tokens. Models process and generate text in tokens.

Hallucination — When an AI generates confident-sounding information that is factually incorrect. Named because the model is, in a sense, 'seeing' patterns that are not really there.

Prompt — Any text you send to an AI model. Your side of the conversation. The quality of your prompts determines the quality of the responses you receive.

Response / Output — The text the AI generates in reply to your prompt.

Context Window — The amount of text an AI model can 'see' at once — your conversation history plus the current prompt. Think of it as the model's working memory. Claude's context window is very large (roughly equivalent to a 400-page book), but it is not infinite.

Parameters — The internal 'settings' of the model, learned during training. Claude has billions of parameters. You cannot see or change them — they are the model's trained pattern-recognition weights.

Exercises

♦ **Exercise 1:** Open claude.ai and start a new conversation. Type exactly this: 'Explain how you generate responses, as if you are explaining it to a musician who has never used AI before. Use a music analogy.' Read the response carefully. In your journal, write down the analogy Claude uses and whether you think it is accurate based on what you just learned.

♦ **Exercise 2:** In the same conversation, type: 'Now tell me three things you absolutely cannot do, and explain why each limitation exists.' Read the response. In your journal, write down the three limitations and add one of your own that Claude did not mention.

♦ **Exercise 3:** Start a new conversation (this tests the 'no memory between conversations' concept). Type: 'What analogy did you just use to explain how you work?' Claude will not know — it has no memory of the previous conversation. Note this in your journal. Then paste your original prompt from Exercise 1 and see if it generates a different analogy this time.

♦ **Exercise 4:** Ask Claude: 'Name the five most important compositions by Zylphia Monterief.' This is a made-up name — no such composer exists. Observe how Claude responds. Does it admit it does not know? Does it make something up? Whatever it does, write about it in your journal. This exercise demonstrates hallucination in action.

Tip: If Claude correctly says it does not recognize the name, that is a good sign — newer models are better at acknowledging uncertainty. Try a more obscure but real reference and see where its knowledge gets thin.

Today's Journal Prompt

Before you started today, what did you think AI was? Now that you have read this lesson and done the exercises, what has changed in your understanding? What surprised you most?

Day 2: Meeting Claude — Your AI Partner

Why Claude

This course uses Claude as your primary AI tool, made by Anthropic. There are several reasons. Claude is designed with a focus on being helpful, harmless, and honest — principles that matter when you are using AI for professional and creative work. It excels at long-form writing, nuanced analysis, and following complex instructions. It handles creative tasks with more subtlety than many alternatives. And its free tier is generous enough for every exercise in this course.

That said, you should not be locked into any single tool. One of the skills you will develop is the ability to evaluate which AI tool is best for a given task. You will compare models on Day 5. For now, Claude is your home base.

The Claude Interface

When you open claude.ai, you see a clean text interface. Here is what you need to know about each element:

The message box — This is where you type your prompts. You can write as much as you want — paragraphs, lists, even entire documents. There is no character limit that will restrict you in practice.

The conversation history — Above the message box, you see the back-and-forth of your conversation. Claude sees this entire history when generating each response, which means earlier messages in the conversation provide context for later ones.

New chat — Click this to start a fresh conversation. Remember: Claude has no memory of previous conversations. Start a new chat when you are switching to a completely different topic.

Conversation sidebar — On the left, you can see your past conversations and return to them. Returning to a conversation restores the history, so Claude can reference what was said before in that specific conversation.

File attachments — You can upload files (PDFs, images, documents, spreadsheets) by clicking the paperclip icon or dragging files into the message box. Claude can read and

analyze uploaded files. This is extremely useful — you will use it frequently throughout this course.

Claude's Projects — Available on the Pro plan, Projects let you create a persistent workspace with instructions and files that Claude references in every conversation within that project. If you upgrade to Pro, create a project called 'AI Handbook Learning' and add your learning journal to it.

Your First Real Conversation

Today you are going to have a genuine working session with Claude. Not just testing it — actually using it to accomplish something useful for your music studies.

Step 1: Start a new conversation. Begin by giving Claude context about who you are. This is crucial — the more Claude knows about you, the more tailored and useful its responses will be.

Prompt: "I am a [your year, e.g., junior] music student studying [your instrument/focus] at [your school]. I am working on [something specific, e.g., preparing for my junior recital, writing a theory paper, learning a new piece]. I would like your help with [specific task]. My level of experience with [the relevant topic] is [beginner/intermediate/advanced]."

Step 2: Read Claude's response. Notice how it tailors its answer to the context you provided. Now go deeper. Ask a follow-up question that builds on what Claude said. This is where the real value of AI emerges — in the back-and-forth dialogue, not in one-off queries.

Step 3: Challenge Claude. If something in its response seems wrong, generic, or unhelpful, say so. You might type: 'That advice is too generic. I need something more specific to [your situation].' Or: 'I do not think that is accurate. Can you double-check?' Claude responds well to pushback — it will refine and improve its response.

Step 4: Ask Claude to change its format. Try: 'Can you reorganize that as a numbered list?' Or: 'Give me that as a week-by-week schedule.' Or: 'Simplify that — explain it like I am a complete beginner.' The ability to request different formats is one of AI's most practical features.

Exercises

♦ **Exercise 1:** Have Claude explain a music theory concept you have always found confusing. Start broad, then ask progressively more specific follow-up questions. See how deep you can go in a single conversation. Aim for at least ten exchanges back and forth.

♦ **Exercise 2:** Ask Claude to generate a one-week practice schedule for your current repertoire. Be very specific: tell it exactly what pieces you are working on, what technical

challenges you face, when your recital or jury is, and how much practice time you have per day. Evaluate the schedule it creates. Modify it by telling Claude what to change.

♦ **Exercise 3:** Upload a file to Claude. This could be a PDF of a syllabus, a photo of sheet music, or a document you are working on. Ask Claude to analyze it, summarize it, or help you with a specific aspect of it. Notice how having the actual document in the conversation changes the quality of Claude's responses versus just describing the document in words.

♦ **Exercise 4:** Have a creative conversation. Ask Claude: 'I want to compose a short piece for [your instrument] that evokes the feeling of [a specific emotion or scene]. Help me brainstorm ideas for the form, harmony, and mood.' Engage with its suggestions. Push back where your artistic instinct disagrees. This is AI as creative collaborator — not replacement.

Today's Journal Prompt

Describe the best moment from today's conversation with Claude — the point where it said something genuinely useful or surprising. What made that moment work? What did you do in your prompt that led to that useful response?

Day 3: The Science of Prompting

Why Prompting Is a Skill

If AI is a musical instrument, prompting is the technique required to play it well. Just as a violinist's tone depends on bow speed, pressure, and contact point — not on the violin itself — the quality of AI output depends on how you construct your prompts. Two people can use the same AI model and get dramatically different results based solely on their prompting skill.

The good news: prompting is a learnable skill, and you are going to learn it today. By the end of this lesson, you will have a reliable framework for writing prompts that consistently produce excellent results.

The Five Elements of an Effective Prompt

Every great prompt contains some or all of these five elements:

1. Role

Tell the AI who it should be in this conversation. This sets the tone, vocabulary, and depth of the response. The more specific the role, the better the output.

Prompt: "You are an experienced jazz piano teacher who specializes in helping classical pianists transition to jazz improvisation."

Compare that to just asking 'How do I learn jazz piano?' The role-assigned version will give you advice that is specifically tailored to someone coming from a classical background, using language and concepts that bridge the two worlds.

More examples of effective roles:

- 'You are a grants administrator who has reviewed thousands of NEA applications and knows what makes them succeed.'
- 'You are a music historian specializing in 20th-century American composition.'
- 'You are a social media strategist who works exclusively with independent musicians.'
- 'You are a concert promoter in a mid-size city who books 200 shows a year.'

2. Context

Give the AI background information about you, your situation, and what you have already done or know. Context is the difference between generic advice and advice that actually fits your life.

Prompt: "I am a graduate violin student preparing for my doctoral recital in three months. My program includes the Bartók Solo Sonata, Brahms Sonata No. 3, and a new commission by a local composer. I practice four hours daily but have been struggling with the rhythmic complexity in the Bartók, specifically the fugue movement."

Now Claude knows your level, your timeline, your specific repertoire, your practice habits, and your specific challenge. Its advice will be dramatically more useful than if you had just asked 'How do I prepare for a recital?'

3. Task

State clearly and specifically what you want the AI to do. Vague tasks produce vague results. Specific tasks produce specific, actionable results.

Weak task: 'Help me with my recital.'

Strong task: 'Create a detailed three-month practice plan that allocates my four daily hours across these three pieces, with specific weekly milestones and technique-building exercises targeting the Bartók fugue rhythms.'

4. Format

Tell the AI how you want the output structured. This is one of the most underused elements of prompting, and one of the most powerful.

- 'Give me this as a table with columns for Week, Piece, Focus Area, and Daily Time Allocation.'
- 'Write this as a bulleted list, no more than ten items.'
- 'Format this as a formal letter addressed to [name].'
- 'Keep your response under 200 words.'

- 'Use headers and subheaders to organize this.'
- 'Write this at a level appropriate for a concert audience with no music training.'

5. Constraints

Tell the AI what to avoid, what limits to respect, and what assumptions not to make. Constraints prevent the AI from going off in unhelpful directions.

- 'Do not assume I have access to a practice room with a piano — I only have my violin.'
- 'Do not suggest I change my repertoire — the program is set.'
- 'Avoid jargon — this needs to be readable by a non-musician donor audience.'
- 'Do not exceed 500 words — this is for a printed program that has limited space.'
- 'Focus only on the harmonic analysis, not the historical context.'

Putting It All Together

Here is what a complete, well-crafted prompt looks like using all five elements:

Prompt: "You are an experienced orchestra audition coach who has helped dozens of musicians win professional positions. [ROLE]

I am a master's-level horn player preparing for my first professional orchestra audition in six weeks. The audition is for a section position with a mid-tier regional orchestra. The required excerpts include Beethoven 7 (all movements), Brahms 2 (first movement), Strauss Till Eulenspiegel, and two excerpts of my choice. I practice three hours daily and currently play in my school's orchestra and a freelance brass quintet. [CONTEXT]

Create a six-week preparation plan with specific daily and weekly goals. Include mental preparation strategies, mock audition scheduling, and suggestions for my two choice excerpts that would show range and musicality. [TASK]

Format this as a week-by-week breakdown with daily bullet points. Include a separate section for mental preparation. [FORMAT]

Do not suggest I take lessons with a new teacher — I am happy with my current teacher. Do not suggest repertoire changes to the required list. Keep each week's plan to one page or less. [CONSTRAINTS]"

You do not need all five elements in every prompt. A quick question might only need a task. But for any substantial request — anything where you want high-quality, tailored output — the more elements you include, the better your results will be.

The Iteration Principle

Even a perfect prompt rarely produces a perfect response on the first try. The real skill is in iteration — refining your request based on what the AI gives you. Think of it like rehearsing with an accompanist: the first run-through reveals what needs adjustment, and you communicate those adjustments clearly.

Common iteration moves:

- 'That is good, but the tone is too formal. Make it more conversational and warm.'
- 'The practice plan is solid, but I need more detail on the Bartók section. Expand that part.'
- 'You included some information that seems inaccurate. The Brahms 3 Sonata is in D minor, not D major. Please correct and regenerate.'
- 'This is too long. Cut it in half while keeping the most important points.'
- 'I like the structure but not the content of section 2. Rewrite just that section with a focus on [specific thing].'
- 'Give me three alternative versions of the opening paragraph. I want to choose the best one.'

Exercises

♦ **Exercise 1:** Write a prompt using all five elements (Role, Context, Task, Format, Constraints) to get Claude to create something useful for your music studies right now. It could be a practice plan, program notes, a theory explanation, or anything else you genuinely need. Save both your prompt and Claude's response in your journal.

♦ **Exercise 2:** Take the same task from Exercise 1 and write the worst possible prompt for it — something vague, contextless, and generic. Send it to Claude and compare the response to what you got from your well-crafted prompt. Document the difference in your journal. This exercise makes the value of good prompting viscerally clear.

♦ **Exercise 3:** Practice iteration. Send Claude a substantial request (at least a paragraph). When you get the response, do not accept it as-is. Iterate at least five times — adjusting tone, length, detail, focus, or format each time. Save the final version and your original prompt. Note how much better the final version is compared to the first response.

♦ **Exercise 4:** Create a 'Prompt Template' for a task you do regularly. For example, if you often need to write program notes, create a template like: 'You are [role]. I am performing [piece] by [composer] on [date] for [audience type]. Write program notes that are [length] and include [specific elements]. Avoid [things to avoid].' Save this template in your journal for reuse.

Today's Journal Prompt

Which of the five prompting elements (Role, Context, Task, Format, Constraints) made the biggest difference in the quality of responses you received today? Why do you think that one element mattered most?

Day 4: Advanced Prompting Techniques

Beyond the Basics

Yesterday you learned the five fundamental elements of a good prompt. Today you learn the techniques that experienced AI users rely on daily. These are not tricks — they are communication strategies that leverage how language models process information.

Technique 1: Few-Shot Prompting (Teaching by Example)

Instead of describing what you want in abstract terms, show the AI an example of the output you are looking for. This is called 'few-shot prompting' because you give the model a few 'shots' (examples) to learn from.

Here is how it works. Say you want Claude to write practice journal entries in a specific style:

Prompt: "I want you to write practice journal entries in the following style. Here is an example of the format and tone I want:

Date: March 15, 2026

Piece: Rachmaninoff Piano Concerto No. 2, Mvt. 1

Time: 90 minutes

Focus: Cadenza passage, mm. 203-245

What worked: Hands-separate practice at half tempo revealed a fingering issue in the left hand arpeggios. Switching from 1-2-4-5 to 1-2-3-5 at m. 228 solved the stumbling.

What needs work: Still rushing the build from m. 230 to the climax. Need to practice with metronome tomorrow, gradually increasing from quarter = 60 to quarter = 88.

Mood: Frustrated at first, then satisfied after the fingering breakthrough.

Now write five practice journal entries for a cellist working on the Dvořák Cello Concerto, spacing them across two weeks. Make each entry specific and detailed, with realistic breakthroughs and challenges."

Because you showed Claude exactly what a good entry looks like, it can replicate the format, tone, level of detail, and type of content. Without the example, you would get something generic. With the example, you get something that matches your vision.

Technique 2: Chain-of-Thought Prompting

When you need the AI to work through a complex problem, ask it to show its reasoning step by step. This is called 'chain-of-thought' prompting, and it dramatically improves the quality of responses for analytical or multi-step tasks.

Prompt: "Analyze the harmonic structure of the first eight bars of Debussy's 'Clair de Lune.' Work through it chord by chord, explaining: (1) what the chord is, (2) its function in the key, (3) what makes Debussy's voice leading distinctive in each transition, and (4) how this relates to Impressionist harmonic language more broadly. Think step by step."

The phrase 'think step by step' or 'work through this step by step' is almost magical in its effect on AI output quality. It forces the model to build its analysis incrementally rather than jumping to conclusions, which produces more accurate, thorough, and educational responses.

Technique 3: Persona Stacking

You can ask Claude to embody multiple perspectives at once, which is incredibly useful for getting well-rounded advice.

Prompt: "I am writing a grant application for a new music composition. I want you to review my draft from three perspectives: (1) as a grant panel reviewer who has read hundreds of applications and knows what makes one stand out, (2) as a professional composer who can evaluate whether my artistic vision is clearly communicated, and (3) as a nonprofit administrator who cares about budget justification and community impact. Give me feedback from each perspective separately, clearly labeled."

Technique 4: Structured Output Requests

You can ask Claude to return information in very specific structures — tables, CSV format, JSON, numbered lists with sub-points, comparison matrices, or any other format you need.

Prompt: "Create a comparison table of five AI music tools (Moises.ai, LANDR, Suno, Descript, and ElevenLabs) with the following columns: Tool Name, What It Does, Best For, Free Tier Limitations, Cost of Paid Tier, My Recommended Use Case as a Music Student."

Tables are particularly useful because they force conciseness and make information easy to scan. Any time you are comparing options or organizing features, request a table.

Technique 5: The 'Before You Begin' Technique

For complex tasks, ask Claude to plan its approach before executing. This catches misunderstandings before they result in wasted output.

Prompt: "I need you to write a 2,000-word article about the impact of AI on music education. Before you begin writing, outline your planned structure: what sections you will include, the main argument of each section, and approximately how many words you will allocate to each. I will review the outline and give you feedback before you write the full article."

This is like a director giving notes on a rehearsal plan before the actual rehearsal. It saves enormous time because adjusting an outline takes seconds, while rewriting 2,000 words takes much longer.

Technique 6: Negative Prompting

Sometimes the best way to get what you want is to specify what you do not want. AI models have certain default patterns — they tend to be comprehensive, formal, and eager to please. Negative prompting counteracts these defaults.

Prompt: "Write a bio for my website. Do NOT use any of these clichés: 'passionate musician,' 'lifelong love of music,' 'explores the boundaries,' 'brings audiences on a journey.' Do NOT write in the third person — use first person. Do NOT exceed 150 words. Do NOT include a list of degrees — weave my education naturally into the narrative."

Technique 7: The Revision Sandwich

When you want to improve existing text (yours or AI-generated), give Claude the original, tell it what to keep, and tell it what to change. This preserves what works while fixing what does not.

Prompt: "Here is my current artist bio [paste bio]. I like the opening sentence and the mention of my chamber music work. But the middle section is too long, the tone shifts awkwardly from personal to formal, and the ending is weak. Rewrite it keeping the opening and the chamber music reference, tightening the middle, maintaining a warm conversational tone throughout, and ending with something memorable."

Exercises

♦ **Exercise 1:** Use few-shot prompting. Find a piece of writing you admire — a program note, a review, a blog post by a musician you respect — and paste it into Claude with the instruction: 'Write something in exactly this style and tone, but about [your topic].' Compare the output to what you would get without the example.

♦ **Exercise 2:** Use chain-of-thought prompting to have Claude analyze a piece of music you are studying. Ask it to work through the harmonic analysis measure by measure, explaining its reasoning at each step. Then ask it to identify the three most interesting harmonic moments and explain why they are effective.

♦ **Exercise 3:** Use persona stacking. Ask Claude to evaluate your current semester schedule or practice routine from three different perspectives: a time management expert, your dream mentor in your field, and a sports psychologist who specializes in performance anxiety.

♦ **Exercise 4:** Use the 'Before You Begin' technique. Give Claude a complex creative task — like writing a series of social media posts for an upcoming recital — and have it outline its plan before executing. Review the plan, make adjustments, and then let it execute. Compare this to what you would get by just saying 'Write social media posts for my recital.'

Today's Journal Prompt

Which advanced technique felt most natural to you? Which do you think you will use most often? Write down one specific scenario from your music life where each technique would be useful.

Day 5: Comparing AI Models

The AI Landscape in 2026

Claude is not the only AI model available to you, and being AI-literate means understanding the landscape well enough to choose the right tool for each task. Think of it like knowing when to use a grand piano versus an upright versus a keyboard — they are all pianos, but each has strengths suited to different situations.

Here are the major AI models you should know about as of April 2026:

Claude (by Anthropic)

Best for — Long-form writing, nuanced analysis, creative collaboration, following complex instructions, handling large documents, coding assistance

Strengths — Excellent at maintaining tone and style across long outputs. Handles ambiguity well. Less likely to produce harmful or misleading content. Reads and analyzes uploaded files exceptionally well. Extended thinking mode for complex reasoning.

Limitations — Cannot browse the internet in real time (though this may change). Cannot generate images natively. Knowledge has a training cutoff date.

Free tier — Generous daily message limit. Access to a capable model. File uploads included.

Pro tier (\$20/month) — Much higher message limits. Access to the most powerful models. Projects for persistent context. Claude Code access.

ChatGPT (by OpenAI)

Best for — General knowledge questions, coding, image generation (via DALL-E), browsing the internet, voice conversations

Strengths — Massive user base means extensive community resources and tutorials. Integrated image generation. Can browse the web for current information. Voice mode for spoken conversations. Plugin ecosystem.

Limitations — Can be verbose and repetitive. Sometimes prioritizes agreeableness over accuracy. Creative writing can feel generic. The free tier is more limited.

Free tier — Access to the current model with daily limits. Image generation limited.

Plus tier (\$20/month) — Higher limits. Access to all features including advanced voice, image generation, and web browsing.

Gemini (by Google)

Best for — Tasks that benefit from Google integration (search, Gmail, Docs), multimodal tasks (text + image + video), research requiring current information

Strengths — Deep integration with Google workspace. Can process video and audio natively. Excellent at research tasks due to search integration. Very large context window.

Limitations — Creative writing quality is generally below Claude and ChatGPT. Can feel more corporate in tone. Privacy considerations due to Google's data practices.

Free tier — Generous access to Gemini Pro. Integrated into Google Search.

Advanced tier (\$20/month) — Access to most powerful Gemini models. Deeper Google workspace integration.

Other Models Worth Knowing

Meta's Llama (free, open source) — Runs locally on your computer with enough hardware. Good for privacy-sensitive tasks. The fact that it is open source means anyone can build on it.

Mistral (French company) — Strong at multilingual tasks. Good balance of capability and efficiency. Available in various sizes.

Perplexity — Not a model itself but an AI-powered search engine that cites its sources. Excellent for research where you need verifiable information.

How to Choose

In practice, here is a simple decision framework:

- Need to write something long, nuanced, or creative → Claude
- Need to search the internet for current information → ChatGPT or Perplexity
- Need to generate an image → ChatGPT (DALL-E) or dedicated image tools (covered in Week 4)
- Need to work within Google Docs/Gmail/Calendar → Gemini
- Need to analyze a large document or PDF → Claude
- Need a quick factual answer → Any of them, but verify with a second source

Exercises

♦ **Exercise 1:** Run the same prompt through Claude, ChatGPT, and Gemini (all free tiers). Use this prompt: 'Write 300-word program notes for a performance of Beethoven's Piano Sonata No. 23 (Appassionata) for an audience of classical music subscribers who are knowledgeable but not professional musicians. Include historical context and listening guidance.' Compare the three responses. In your journal, note which you prefer and why.

♦ **Exercise 2:** Test each model's ability to handle a music-specific knowledge task. Ask all three: 'Explain the difference between sonata form and sonata-rondo form, with specific examples from the standard repertoire for [your instrument]. Include a comparison table.' Note which model gives the most accurate and useful response.

♦ **Exercise 3:** Test creativity. Ask all three: 'I am a [your instrument] player who wants to commission a new 5-minute solo work. Help me write a creative brief for a composer, describing the kind of piece I envision — the mood, the technical elements I want to showcase, the audience I imagine, and any non-musical inspirations. Make it vivid and specific.' Which model produces the most compelling creative vision?

♦ **Exercise 4:** Sign up for Perplexity (perplexity.ai, free) and test it as a research tool. Ask it: 'What are the most significant grants available to emerging classical musicians in the United States in 2026, with application deadlines and award amounts?' Compare its sourced, cited response to what Claude gives you for the same question. Note the difference between generated knowledge and searched knowledge.

Today's Journal Prompt

Based on today's comparisons, which AI model do you see yourself using most? Did any of the results surprise you — was one model much better or worse than you expected? How does having multiple AI options change how you think about using them?

Day 6: AI Ethics, Bias & Limitations

Why This Matters for Musicians

AI is not neutral. It was built by humans, trained on human-created data, and reflects the biases, blind spots, and values embedded in that data and those decisions. As a musician — an artist whose work shapes culture — you have a responsibility to understand these issues before integrating AI into your creative and professional life.

This is not a theoretical exercise. The ethical questions about AI in music are immediate and practical: Is it okay to use AI to help write grant applications? Where is the line between AI-assisted composition and AI-generated composition? If AI masters your recording, who is the producer? When AI generates your social media content, is that authentic? These questions do not have universal answers, but you need to have your own thoughtful positions.

Bias in AI

AI models are trained on existing text from the internet, books, and other sources. This training data reflects the world as it has been documented — which means it reflects the biases of who has had the power and platform to create that documentation.

For musicians, this shows up in specific ways:

Western classical music dominance — Ask most AI models about 'great music' or 'important composers' and you will get a heavily Western European, predominantly male, predominantly white list. This is not because AI has evaluated all the world's music and concluded Western classical is best — it is because Western classical music is disproportionately represented in the training data.

Genre hierarchies — AI models tend to treat classical and jazz as 'serious' music and pop, hip-hop, country, and electronic music as 'less serious' — reflecting long-standing cultural biases in music criticism and academia, not any objective truth about artistic merit.

Gender and representation — Ask AI to list 'great violinists' and notice the gender ratio. Ask it about 'great conductors' and notice it even more. AI reproduces the historical underrepresentation of women, people of color, and non-Western artists in its outputs because that underrepresentation exists in its training data.

Contemporary vs. historical — AI models know much more about historical music (which has been written about extensively) than about contemporary music (which has less written documentation in the training data). This means AI advice about current industry practices, living composers, or emerging trends is less reliable than its knowledge of established canon.

Hallucination: When AI Makes Things Up

As you learned on Day 1, AI can generate confident-sounding information that is completely wrong. This is especially dangerous in music contexts where you might not easily verify the information:

- AI might attribute a composition to the wrong composer
- It might describe a piece's structure incorrectly
- It might cite music theory 'rules' that are actually just common conventions
- It might invent historical facts — dates, events, quotes — that never happened
- It might recommend resources (books, recordings, websites) that do not exist

The rule: never trust AI for factual claims without verification, especially when the information will be published, performed, or shared with others. If Claude tells you that a Beethoven sonata was composed in 1803, check it. If Claude says a music theory rule applies, verify it against a textbook. AI is a starting point for research, not a replacement for it.

The Ethics of AI-Generated Creative Work

This is the big one for musicians. Where is the line between using AI as a tool and passing off AI's work as your own?

Consider these scenarios and where you would draw the line:

- Using AI to brainstorm ideas for a composition, then composing the music yourself — most people consider this ethical, similar to being inspired by a conversation with a colleague
- Using AI to generate chord progressions that you then arrange, orchestrate, and develop — a gray area that depends on how much transformation you add
- Using AI to write your program notes entirely — generally acceptable if you review, edit, and approve the output, especially since program notes are informational rather than creative expression
- Using AI to generate a melody or complete musical passage that you perform as your own composition — most of the music community currently considers this problematic, equivalent to uncredited ghostwriting
- Using AI-generated music as background or underscoring in a multimedia project — increasingly common and generally accepted with disclosure
- Submitting AI-generated work for academic credit without disclosure — universally considered academic dishonesty

There is no single right answer to where the line falls, and the norms are evolving rapidly. What matters is that you think about these questions deliberately, form your own ethical framework, and are transparent about your use of AI. Disclosure is almost always the right choice — saying 'I used AI to help draft this' or 'AI-assisted arrangement' is honest and, increasingly, unremarkable.

AI and Copyright

As of 2026, the legal landscape around AI and copyright is still developing, but here are the key principles you should know:

- AI-generated content (text, images, music) is generally not copyrightable by itself in the United States — because copyright requires human authorship
- If you substantially modify, arrange, or curate AI output, your modifications may be copyrightable
- Using AI tools does not give you permission to reproduce copyrighted material. If an AI generates something that closely resembles a copyrighted work, using that output could still be infringement
- AI music generation tools (Suno, Udio) are currently facing lawsuits from major record labels over whether their training on copyrighted music constitutes fair use
- When in doubt, treat AI output as a starting point that you transform into something original, rather than as a finished product you claim wholesale

Exercises

♦ **Exercise 1:** Test for bias. Ask Claude: 'Name the ten greatest composers in music history.' Then ask: 'Now name ten composers from that same era who are equally important but are underrepresented due to gender, race, or geography.' Compare the two lists. Ask Claude to explain why the first list looked the way it did. Write about what you observe.

♦ **Exercise 2:** Test for hallucination. Ask Claude detailed factual questions about music you know well — your instrument's repertoire, your school's history, a specific performer's discography. How accurate is it? Where does it get things wrong? Document every error you find.

♦ **Exercise 3:** Write your personal AI ethics statement. In 300-500 words, describe your principles for using AI in your creative work, your academic work, and your professional communications. Where will you draw the lines? When will you disclose AI use? Save this — you will refine it throughout the course.

♦ **Exercise 4:** Research one current AI ethics debate in music. Use Perplexity or Google to find a recent article about AI copyright lawsuits, AI music generation controversies, or AI's impact on music industry employment. Read it, then ask Claude to help you understand the arguments on both sides. Form your own opinion and write it in your journal.

Today's Journal Prompt

What is the most important ethical principle for musicians using AI? What would you tell a fellow student who asked 'Is it cheating to use AI?'

Day 7: Review, Practice & Reflection

Week 1 Summary

This week you have learned what AI actually is (a pattern predictor, not a thinker), how to communicate with it effectively (the five elements of prompting), advanced techniques that dramatically improve output quality (few-shot, chain-of-thought, persona stacking, and more), how different AI models compare, and the ethical considerations that should guide your use of AI.

You should now be able to:

- Explain to someone else how AI works, without mystification or hype
- Write prompts that consistently produce useful, tailored output
- Use iteration to refine AI responses until they meet your standards
- Choose the right AI model for different tasks
- Think critically about AI output — checking for bias, hallucination, and ethical issues

Review Exercises

♦ **Prompt Challenge 1:** Write the best possible prompt to get Claude to create a detailed listening guide for a piece you are currently studying. Use Role, Context, Task, Format, and Constraints. Iterate at least three times. The final product should be something you would genuinely share with a friend.

♦ **Prompt Challenge 2:** Use chain-of-thought prompting to have Claude walk through a complex musical analysis — perhaps the form of a sonata movement, the harmonic language of a jazz standard, or the orchestration choices in a symphonic work. The analysis should teach you something new.

♦ **Prompt Challenge 3:** Write a prompt that produces a one-page document you can actually use this week — a practice schedule, a draft email to a professor, a set of audition warm-ups, or a summary of a reading assignment. The goal is to demonstrate that AI is already a useful daily tool.

Anthropic Academy

If you have not already, this is a good day to start the free courses on Anthropic's Academy (anthropic.skilljar.com). Two courses are especially relevant at this point:

- Claude 101 — Covers core features and practical prompting. Reinforces everything you learned this week with slightly different framing and additional examples.
- AI Fluency: Framework & Foundations — A broader look at AI fluency using Anthropic's 4D framework. Gives you vocabulary and concepts for thinking about AI more systematically.

Both are free, take 2-4 hours each, and award certificates you can add to LinkedIn. They fit well into the one-hour-a-day framework if you spread them across this day and next week's open time.

This Week's Journal Prompt

Look back at your Day 1 journal entry about what you thought AI was before starting. How has your understanding changed? What do you know now that you did not know seven days ago? What are you most excited to learn next?

Week 2: AI for Writing, Research & Communication

You now know how to talk to AI effectively. This week, you put that skill to work on the writing tasks that fill a musician's professional life. By Sunday, you will have a portfolio of AI-assisted writing: program notes, a professional bio, grant language, emails, social media content, and more — all in your own voice, all ready to use.

Day 1: Writing Program Notes, Bios & Artist Statements

Program Notes

Every performer writes program notes. They are the audience's guide to what they are about to hear — context that transforms passive listening into active engagement. Writing good program notes is an art: they need to be informative without being academic, personal without being self-indulgent, and engaging without being condescending.

AI is exceptionally good at helping with program notes because the format is well-defined and the source material (music history, theory, composer biography) is well-represented in training data. But you need to guide AI carefully to avoid the two most common failures: notes that read like a Wikipedia article (too dry) or notes that read like a press release (too breathless).

Step-by-Step: Writing Program Notes with AI

Step 1: Start with your own knowledge. Before you even open Claude, write down in plain language: what is this piece, why did you choose it, what should the audience listen for, and what do you find personally meaningful about it. This does not need to be polished — it is raw material. Even just bullet points.

Step 2: Build your prompt. Include your raw notes, the specific piece, the performance context, and your audience.

Prompt: "You are a music writer who specializes in concert program notes — the kind you find in programs at professional chamber music concerts. Your writing style is warm, intelligent, and conversational — you treat the audience as smart people who may not have formal music training.

I am performing Brahms's Clarinet Sonata No. 1 in F minor, Op. 120/1, on my senior recital in May. My audience will be a mix of music students, faculty, family, and friends — some knowledgeable, some not.

Here are my personal notes about the piece: [paste your raw notes here – what you find compelling, why you chose it, what the audience should listen for]

Write program notes of approximately 400 words. Include brief historical context (Brahms's late period, his friendship with clarinetist Richard Mühlfeld), a movement-by-movement listening guide that highlights specific moments the audience should listen for, and weave in my personal perspective. The tone should feel like a knowledgeable friend explaining the piece over coffee, not like a textbook or a press kit."

Step 3: Review critically. When Claude generates the notes, read them carefully. Check for factual accuracy (especially dates and biographical details). Check that the tone matches what you want. Check that your personal perspective came through, not just generic background information.

Step 4: Iterate. Common adjustments you might request:

- 'The historical background is too long. Cut it in half and expand the listening guide.'
- 'This sounds too academic. Make it warmer and more personal.'
- 'You did not include the point I made about [specific thing]. Work that in.'
- 'The transition between the historical context and the listening guide is awkward. Smooth it out.'
- 'Add a sentence at the end that connects back to my personal connection to the piece.'

Step 5: Make it yours. Copy the final version into a document and edit it by hand. Change words to your vocabulary. Adjust phrases to match how you actually talk. This final pass is what transforms AI-assisted writing into your writing. It should sound like you.

Artist Bios

Every musician needs a bio in at least three lengths: a long version (300-500 words, for your website and press kits), a medium version (150 words, for programs and introductions), and a short version (50 words, for social media and quick references). AI can help you create all three from a single conversation.

Prompt: "I need to write my professional musician bio. Here is the raw information:

Name: [your name]

Instrument/Focus: [your instrument or area]

Education: [degrees, schools, notable teachers]

Performance highlights: [notable performances, venues, ensembles]

Awards/competitions: [if any]

Current position/activities: [what you are doing now]

Something personal that makes you interesting: [a non-music detail that reveals personality]

Write three versions: long (350 words), medium (150 words), and short (50 words). Write in the third person (standard for musician bios). Make the tone confident but not arrogant, professional but not sterile. Avoid these clichés: 'passionate musician,' 'lifelong dedication,' 'captivating audiences.' Start with something more interesting than my name — lead with what makes me distinctive."

Artist Statements

An artist statement is a first-person document that explains your artistic identity, values, and vision. Unlike a bio (which is factual), an artist statement is philosophical. It answers the question: 'Why do you make the music you make?'

This is one area where AI should assist, not lead. Your artist statement must be authentically yours. Use AI to help you articulate ideas you already have, not to generate ideas for you.

Prompt: "I want to write an artist statement but I am struggling to articulate my ideas. I am going to tell you about my musical values and what drives my work. Ask me questions — one at a time — to help me uncover and articulate my artistic identity. After five questions, use my answers to draft a 250-word artist statement in my voice."

This 'interview' approach is powerful because it mirrors how the best writer collaborators work — they draw out what you already know but have not yet expressed.

Exercises

♦ **Exercise 1:** Write program notes for a piece on your current recital or concert program. Go through all five steps: write your raw notes, craft a detailed prompt, review Claude's output critically, iterate at least three times, and make a final editing pass by hand. Save the finished product.

♦ **Exercise 2:** Create your three-length bio set (long, medium, short). If you already have a bio, give it to Claude and ask for a critical review: 'Read my current bio and tell me honestly what is working and what is not. Then rewrite it.' Compare the original and the rewrite.

♦ **Exercise 3:** Use the interview technique to draft an artist statement. Let Claude ask you questions, answer them honestly, and see what it produces. Then edit the result until every sentence feels genuinely yours.

Day 2: Research with AI — Music History & Theory

AI as Research Assistant

AI is not a replacement for real research — it cannot access library databases, read scores, or listen to recordings. But it is an extraordinarily useful research assistant: it can help you understand complex topics, identify what to look for, generate outlines, explain theory concepts, contextualize your findings, and draft prose that you then verify and refine.

The key mindset: AI generates first drafts of understanding. You verify, deepen, and own the final understanding.

Using AI for Music History Research

Here are the most effective patterns for music history research with AI:

Pattern 1: The Contextualizer

Prompt: "I am studying [piece] by [composer]. Give me the historical context I need to understand why this piece matters: what was happening in the composer's life when they wrote it, what was happening culturally and politically, what musical traditions it builds on, and what influence it had on what came after. Be specific — I want dates, names, and concrete details, not vague generalities."

Tip: Always verify specific dates and factual claims against a reliable source like Grove Music Online or a published biography.

Pattern 2: The Theory Explainer

Prompt: "Explain [theory concept] as if you are tutoring me one-on-one. Start with the simplest version of the concept, then gradually add complexity. Use specific examples from the standard repertoire. When you use technical terms, define them in context. After explaining, give me three practice problems to test my understanding, then check my answers."

Pattern 3: The Comparison Engine

Prompt: "Compare [piece A] and [piece B] across these dimensions: form, harmonic language, treatment of [instrument], historical significance, and performance challenges. Present this as a detailed comparison table followed by a paragraph synthesis of the most interesting differences."

Pattern 4: The Source Finder

Prompt: "I am writing a paper on [topic]. Suggest ten scholarly sources I should look for — specific books, journal articles, and dissertations that would be relevant. For each, tell me what it covers and why it would be useful for my paper. Include a mix of foundational texts and recent scholarship. Note: I will verify that these sources exist before I cite them."

Tip: AI often generates plausible-sounding but nonexistent citations. Always verify every source in a library database before using it.

Exercises

- ♦ **Exercise 1:** Use the Contextualizer pattern for a piece you are currently studying. Read Claude's response, then verify at least three specific factual claims using Grove Music Online, Wikipedia, or a published book. Note any errors in your journal.
- ♦ **Exercise 2:** Use the Theory Explainer pattern for a concept from your current theory class that you find challenging. Go back and forth with Claude until the concept clicks. If you can explain it in your own words without looking at Claude's explanation, you have learned it.
- ♦ **Exercise 3:** Use the Source Finder pattern for a paper topic. Then go to your school library's database and check how many of the suggested sources actually exist. What percentage were real? For the real ones, do Claude's descriptions match what the sources actually cover?

Day 3: Academic Writing — Papers, Theses & Applications

The Right Way to Use AI for Academic Work

Let's address this directly: using AI to write your papers for you is academic dishonesty at virtually every institution. That is not what this lesson teaches. Instead, you will learn how to use AI as a legitimate academic tool — the way you would use a writing tutor, a research librarian, or a knowledgeable colleague who helps you think through your ideas.

Legitimate uses of AI in academic work:

- Brainstorming and outlining: 'Help me organize my thoughts on [topic] into a thesis and outline'
- Understanding sources: 'I'm reading this passage from [scholar]. Help me understand what they're arguing'
- Getting unstuck: 'I'm stuck on the transition between my analysis of [X] and my discussion of [Y]. Suggest three possible ways to connect them'

- Editing and revision: 'Read my draft and identify weaknesses in my argument, unclear passages, and grammatical errors'
- Formatting: 'Convert these citations to Chicago/MLA/APA format'

Illegitimate uses:

- Having AI write your paper or substantial portions of it without disclosure
- Having AI generate analysis and presenting it as your own thinking
- Submitting AI-generated text without your professor's knowledge and permission

When in doubt, ask your professor. Many professors are developing AI policies in real time, and the answer may surprise you — some welcome AI assistance if it is disclosed, while others prohibit it entirely. Know your institution's policy.

Using AI as a Writing Tutor

Prompt: "I am writing an academic paper on [topic] for my [course name] class. Here is my current draft of the introduction: [paste introduction]"

Act as a writing tutor. Do not rewrite this for me. Instead: (1) Identify the thesis statement and evaluate whether it is clear and arguable. (2) Point out any logical gaps or unsupported claims. (3) Note any sentences that are unclear or could be more precise. (4) Suggest what the reader expects to see next based on this introduction. Give me specific, actionable feedback — not just 'this is good' or 'make this better.'"

This approach preserves your intellectual ownership while leveraging AI's ability to provide immediate, detailed feedback — something you might otherwise wait days to get from a professor or writing center.

Graduate School and Fellowship Applications

Applications for graduate schools, residencies, fellowships, and competitions all require written statements. AI can help you draft and refine these, but the content must be authentically yours.

Prompt: "I am applying to [program/fellowship/residency]. The prompt asks me to write about [the specific prompt]. Here is my rough draft: [paste draft]"

Help me strengthen this statement. Specifically: (1) Does my opening grab attention? (2) Is my narrative arc clear — do I tell a story that connects my past, present, and future? (3) Am I specific enough — do I mention concrete experiences, pieces, teachers, and moments, or is it too vague? (4) Does it communicate what makes me distinctive? (5) Is the writing clean, or are there passages that are wordy, clichéd, or unclear?"

Give me detailed feedback, then offer a revised version that addresses the issues you found."

Exercises

♦ **Exercise 1:** Take a paper you have already written (or are currently working on) and use the Writing Tutor prompt to get feedback. Implement the feedback yourself — do not have Claude rewrite it for you. Note how the feedback compares to what you have received from professors.

♦ **Exercise 2:** Use AI to help you outline a paper on a topic from your current coursework. Start by telling Claude your thesis and main points. Have it help you organize them into a logical structure. Then write the paper yourself using the outline as your guide.

♦ **Exercise 3:** If you have an upcoming application, use the application prompt template to get feedback on your statement. If you do not have one coming up, write a draft personal statement for a hypothetical program you would like to attend someday, and use Claude to refine it.

Day 4: Professional Communication — Emails, Proposals & Contracts

The Professional Email

Musicians send more professional emails than they might realize — to professors, venues, collaborators, agents, conductors, contractors, and presenters. A well-written email can open doors. A poorly written one can close them. AI can help you strike the right tone and structure for any professional situation.

Email Templates for Musicians

Here are specific prompt templates for the emails musicians most commonly need to send:

Cold Outreach to a Venue

Prompt: "Write a professional email from me (a [your description] musician) to the booking manager of [venue name/type]. I want to propose a [type of event — concert, residency, series]. I have [relevant credentials]. The email should be concise (under 200 words), professional but warm, and end with a clear call to action. Do not be overly formal or sycophantic. Suggest a subject line that would actually get opened."

Following Up After a Performance

Prompt: "Write a brief thank-you email to [person/organization] after [event]. Mention [specific positive moment from the event]. Express interest in future collaboration without being pushy. Keep it under 150 words."

Proposing a Collaboration

Prompt: "Write an email to [musician/organization] proposing a collaboration on [project]. Explain why I think we would work well together and what I bring to the project. Be specific about the vision but flexible about the details. Professional tone, not too formal. Under 250 words."

Proposals and Pitch Documents

Beyond individual emails, musicians often need to write proposals — for concert series, educational programs, residencies, or collaborations. AI excels at structuring these documents.

Prompt: "I want to propose a [concert series / educational program / residency / collaboration] to [organization]. Help me create a one-page proposal that includes: (1) an attention-grabbing title and subtitle, (2) a compelling two-paragraph summary of the concept, (3) what makes this unique, (4) a practical overview of logistics and timeline, (5) the intended audience and expected impact, and (6) my qualifications to lead this. Here are the details: [provide all relevant information]."

Understanding Contracts

Musicians encounter contracts regularly — performance agreements, recording contracts, teaching contracts, commission agreements. AI can help you understand contract language, though it should never replace legal advice for high-stakes situations.

Prompt: "I received this contract for [type of engagement]. I am going to paste the full text. Please: (1) summarize the key terms in plain English, (2) flag any clauses I should be concerned about or negotiate on, (3) identify anything important that is missing (like cancellation policies, payment timeline, or rights retention), and (4) suggest questions I should ask before signing. Note: I will consult a lawyer for anything high-stakes, but I want to understand the basics first."

Tip: Never rely solely on AI for contract interpretation. For anything involving significant money, rights assignment, or long-term obligations, consult a music industry lawyer. AI helps you understand what you are reading so you can have a more informed conversation with legal counsel.

Exercises

♦ **Exercise 1:** Write three professional emails you actually need to send. Use AI to draft them, then edit them to sound like you, and send them. Real-world application is the best exercise.

♦ **Exercise 2:** Draft a proposal for something you have been wanting to pitch — a concert at a local venue, a workshop at your school, a collaboration with another musician. Use AI to help structure and polish it. If it comes out well, actually send it.

♦ **Exercise 3:** If you have a contract you need to review, use the contract analysis prompt. If not, find a sample performance contract online (many musician organizations publish templates) and practice analyzing it with AI.

Day 5: Social Media & Marketing Content

Why Social Media Matters for Musicians

Whether you find social media inspiring or exhausting, having an online presence is a professional necessity in 2026. It is how audiences find you, how presenters evaluate you, how colleagues connect with you, and how opportunities emerge. The good news: AI can handle the most time-consuming part — generating consistent, quality content — so you can spend your time making music.

Content Strategy: What to Post

Before you create content, you need a strategy. Content without strategy is noise. Ask Claude to help you develop one:

Prompt: "I am a [your description] musician building my social media presence. My goals are [e.g., building an audience for my concerts, connecting with collaborators, establishing expertise in my niche, booking more gigs]. My target audience is [describe]. I can realistically post [number] times per week.

Create a content strategy with: (1) my content pillars — 3-4 recurring themes/categories I should rotate between, (2) a sample two-week content calendar with specific post ideas for each day, (3) best practices for [Instagram / TikTok / YouTube / LinkedIn — pick the platforms you use], and (4) engagement strategies — how to grow beyond just posting."

Content Creation Prompts

Once you have a strategy, here are prompts for specific content types:

Concert/Event Promotion

Prompt: "Write five different social media posts promoting my upcoming [event] on [date] at [venue]. Each post should take a different angle: (1) the music being performed, (2) a personal story about why this concert matters to me, (3) a 'behind the scenes' rehearsal teaser, (4) practical details with a compelling hook, (5) a last-minute urgency post. Each should be under 200 characters for Instagram and include relevant hashtag suggestions."

Educational Content

Prompt: "I want to create educational social media content about [music topic]. Write a series of five posts that each teach one small concept in an accessible, engaging way. These are for a general audience — no jargon, no assumed knowledge. Each should be standalone (people may see them out of order) and end with a question to encourage engagement."

Personal Brand Posts

Prompt: "Write a reflective social media post about [a recent experience, a practice breakthrough, a meaningful performance, a lesson learned]. It should be authentic and personal — not polished PR-speak. Under 300 words. The tone should match my brand: [describe your tone — thoughtful, humorous, earnest, irreverent, etc.]."

Exercises

♦ **Exercise 1:** Develop your content strategy using the prompt above. Then create one week's worth of content — actually write (with AI's help) all the posts you would need for one full week. Edit them to sound like you.

♦ **Exercise 2:** Create a bank of 20 social media post ideas organized by your content pillars. These are ideas you can draw from whenever you need to post, so you never stare at a blank screen.

♦ **Exercise 3:** Draft a social media bio for each platform you use. Your bio is often the first thing people see — make it count. Use AI to generate options, then pick and refine the best one.

Day 6: Multi-Step Workflows & Chain-of-Thought

From Single Prompts to Workflows

Everything you have done so far this week involved individual prompts — one request, one response. But the real power of AI emerges when you chain multiple steps together

into workflows. A workflow is a sequence of prompts where the output of one step becomes the input for the next.

Here are three powerful workflows for musicians:

Workflow 1: The Content Repurposing Pipeline

You write one substantial piece of content (a blog post, program notes, a newsletter), then use AI to transform it into content for every platform you use.

Step 1: Write (with AI's help) a 500-word blog post about an upcoming concert.

Step 2: 'Turn this blog post into a series of 5 Instagram captions, each highlighting a different aspect.'

Step 3: 'Now create a 280-character tweet thread (5 tweets) based on the same content.'

Step 4: 'Write a 100-word email newsletter teaser that links to the full blog post.'

Step 5: 'Create a 60-second spoken script I could use for an Instagram Reel or TikTok.'

One hour of work produces content for every platform for the entire week. This is the efficiency that AI makes possible.

Workflow 2: The Grant Application Pipeline

Step 1: Upload the grant guidelines to Claude. Ask: 'Summarize the key requirements, evaluation criteria, and what reviewers are looking for.'

Step 2: 'Based on these requirements, help me outline my application. Here is my project idea: [describe it].'

Step 3: 'Draft the narrative section based on our outline.' (Then review and edit.)

Step 4: 'Review my budget justification for completeness and clarity: [paste budget].'

Step 5: 'Read the complete draft and evaluate it against the guidelines. Score it on each criterion and suggest improvements.'

Workflow 3: The Performance Preparation Pipeline

Step 1: 'Give me the historical context for [piece].'

Step 2: 'Now give me a detailed structural analysis.'

Step 3: 'Based on the structure and context, suggest an interpretive approach — what are the key expressive decisions a performer needs to make?'

Step 4: 'Create a week-by-week practice plan that moves from technical preparation through interpretive refinement to performance readiness.'

Step 5: 'Write program notes that incorporate the historical context and highlight the interpretive choices I plan to make.'

Exercises

♦ **Exercise 1:** Execute the Content Repurposing Pipeline with a real piece of content for an upcoming event or project. Go through all five steps. Save all the outputs.

♦ **Exercise 2:** Design your own workflow for a task you do regularly. Map out the steps, write the prompts for each step, and test it. Document the workflow in your journal so you can reuse it.

Day 7: Review & Portfolio Assembly

Today you assemble everything you created this week into a usable portfolio of writing. By the end of today, you should have:

- Program notes for at least one piece
- A professional bio in three lengths
- An artist statement draft
- At least three professional emails (sent or ready to send)
- A social media content strategy and one week of content
- At least one multi-step workflow documented for reuse

Create a folder (physical or digital) called 'Professional Writing.' Organize all of this week's outputs into it. This is the beginning of your professional toolkit — a collection of polished, ready-to-use materials that you will continue building throughout the course and your career.

♦ **Portfolio Review:** Open each document you created this week and give it one final editing pass. Read it out loud — hearing the words helps you catch awkward phrasing that your eyes skip over. Fix anything that does not sound like you. When you are done, each piece should be something you would confidently share with a colleague, professor, or prospective employer.

Week 3: AI Audio & Music Creation

This is the week many music students have been waiting for. You are going to explore the AI tools that work directly with audio and music — separating stems, generating music, mastering tracks, transcribing performances, synthesizing voices, and editing audio. By Sunday, you will know what each tool can do, what it cannot do, and how to integrate the useful ones into your musical life.

Day 1: AI Stem Separation — Deconstructing Music

What Stem Separation Is

Stem separation is the ability to take a mixed audio recording — a complete song or performance — and isolate individual elements: vocals, drums, bass, piano, guitar, strings, and more. Five years ago, this required expensive studio software and hours of manual work. Today, AI does it in seconds, and the results are remarkably good.

For musicians, this is a game-changer. Here is why:

- **Practice tool:** Isolate your instrument's part from a recording to hear what the professional player is doing in detail. Or remove your instrument's part to practice along with the rest of the ensemble.
- **Transcription aid:** Isolate a complex inner voice or bass line that is buried in the mix, making it easier to transcribe by ear.
- **Arranging:** Extract individual parts from a recording to study orchestration and arrangement techniques.
- **Teaching:** Create play-along tracks by removing the student's instrument from a recording.
- **Remix and creative work:** Use isolated stems as raw material for creative projects, remixes, or multimedia work.
- **Analysis:** Hear how individual instruments contribute to the overall texture and balance of a recording.

The Tools

Moises.ai

Moises is the most popular AI stem separation tool for musicians, and for good reason. It has a clean interface, works in your browser, and produces good results. Here is exactly how to use it:

Step 1: Go to moises.ai and create a free account. The free tier allows you to process a limited number of tracks per month (typically 3-5), which is enough for learning.

Step 2: Upload a track. You can upload an audio file from your computer (MP3, WAV, FLAC) or paste a YouTube URL. Start with a song you know well — familiarity helps you evaluate the separation quality.

Step 3: Choose your separation type. Moises offers several modes: 2-stem (vocals vs. accompaniment), 4-stem (vocals, drums, bass, other), and 5-stem (vocals, drums, bass, piano, other). For learning purposes, start with 4-stem.

Step 4: Wait for processing (usually 30-60 seconds). When it finishes, you will see individual controls for each stem. You can solo each one, mute it, or adjust its volume.

Step 5: Listen critically. Solo each stem and listen. Notice: How clean is the separation? Can you hear artifacts (distortion, ghosting of other instruments)? How does the quality compare to what you expected?

Step 6: Download the stems you want. Each stem exports as a separate audio file.

LALAL.AI

LALAL.AI is a competing stem separation tool that often produces slightly different results. It is worth trying both on the same track and comparing. Go to lalal.ai, create a free account, and upload the same track you used in Moises. Compare the stem quality. Different tools handle different genres and instruments better — having both in your toolkit means you can choose the better result for each project.

Practical Applications Deep Dive

Application 1: Learning from Recordings

Upload a professional recording of a piece you are learning. Isolate the part for your instrument. Listen on repeat, focusing on: phrasing decisions, rubato, dynamics, articulation, and timing. This is like having a magnifying glass for performance practice.

Then mute your instrument's part and play along with the remaining stems. This is instant, high-quality play-along practice with a 'real' ensemble — not a MIDI backing track.

Application 2: Ear Training

Upload a complex jazz recording. Isolate the bass line. Can you hear the changes more clearly now? Transcribe the bass line, then check your work against the original mix. This isolation-then-verification approach accelerates ear training dramatically.

Application 3: Creating Teaching Materials

If you teach, stem separation lets you create custom practice tracks instantly. Need a play-along for a jazz standard without the piano comping? Need a vocal-only track for a choir rehearsal? Need a drums-only track for a rhythm analysis class? Stem separation creates all of these in seconds.

Exercises

♦ **Exercise 1:** Go to [Moises.ai](https://moises.ai), sign up, and upload a recording of a piece you are currently studying. Separate it into stems. Listen to each stem individually for at least 60 seconds. In your journal, write down three things you noticed in the individual stems that you could not hear in the full mix.

♦ **Exercise 2:** Create a play-along track. Remove your instrument from the mix (or the closest equivalent) and practice along with the remaining stems. How does this compare to practicing with a traditional backing track or metronome?

♦ **Exercise 3:** Upload the same track to both [Moises.ai](https://moises.ai) and [LALAL.AI](https://lalal.ai). Compare the stem quality. Which tool produced cleaner separation for this particular track? Note: results may differ for different genres and instruments.

Day 2: AI Music Generation — Capabilities, Tools & Ethics

The State of AI Music Generation

This is the most contentious area of AI in music, and you need to understand it clearly — both the technology and the debate around it. AI music generation tools can now produce complete songs — with lyrics, vocals, multiple instruments, mixing, and mastering — from a text description. The results range from impressive to uncanny.

But 'can' is not the same as 'should,' and understanding the difference is essential for any serious musician.

The Major Tools

Suno

Suno (suno.com) generates complete songs from text prompts. You describe the style, mood, lyrics, and instrumentation, and Suno produces a fully mixed, fully produced track in about 30 seconds. The free tier gives you a limited number of generations per day.

What to try: Go to suno.com, create a free account, and type a prompt like: 'A gentle acoustic folk song about missing home, fingerpicked guitar, warm male vocal, key of G, moderate tempo.' Listen to what it generates. Notice the structure, the production quality, and the musical choices.

Udio

Udio (ud.io) is similar to Suno but often produces different stylistic results. Its strength tends to be in certain genres (indie, electronic, pop) while Suno may do better in others. Try the same prompt in both tools and compare.

What AI Music Generation Can and Cannot Do (2026)

What it can do well:

- Generate convincing pop, rock, folk, electronic, and hip-hop tracks
- Create backing tracks and mood music for multimedia projects
- Produce music in specific styles with surprising fidelity
- Generate multiple variations quickly for creative brainstorming
- Create scratch tracks and demos to communicate musical ideas

What it cannot do well (yet):

- Complex classical music with proper voice leading and formal development
- Nuanced jazz improvisation with real harmonic sophistication
- Maintain musical coherence over longer forms (beyond about 3 minutes)
- Capture the subtle expressive timing and dynamics of a live performance
- Anything requiring genuine musical understanding — it mimics, it does not comprehend

The Ethics Debate — In Full

The debate around AI music generation has several dimensions:

The Copyright Question

Tools like Suno and Udio were trained on vast libraries of existing music — almost certainly including copyrighted recordings. Major record labels (Universal, Sony, Warner) have filed lawsuits arguing that this training constitutes copyright infringement. The AI companies argue it is fair use, similar to how a human musician learns by listening to existing music. As of mid-2026, these cases are still being decided, and the outcome will shape the industry for decades.

The Labor Question

If AI can generate a decent pop song in 30 seconds, what happens to the session musicians, producers, songwriters, and engineers who currently make their living creating that music? This is not hypothetical — background music for videos, podcasts, advertisements, and games is already being replaced by AI-generated content in many contexts.

The Artistic Question

Does AI-generated music count as art? Is using an AI music tool creative, or is it more like ordering from a menu? And does the answer change depending on how you use it — generating a final product versus using AI output as raw material for further human creative work?

Your Stance

You do not need to resolve these debates today, but you do need to think about them. As a musician, you have more at stake than most people. Your position should be informed by actually trying the tools (so you know what they can do), understanding the arguments on all sides, and reflecting on your own values about music, labor, and creativity.

Exercises

♦ **Exercise 1:** Generate a track on Suno and one on Udio using the same prompt. Listen critically to both. In your journal, write about: What is impressive? What falls short? How does this music compare to music made by human musicians you admire?

♦ **Exercise 2:** Try generating music in your specific genre or instrument's tradition. If you are a classical pianist, try 'a Chopin-style nocturne for solo piano.' If you are a jazz musician, try 'a bebop quartet playing a blues in F.' How well does AI handle your area of expertise? What errors or limitations do you notice that a non-expert might miss?

♦ **Exercise 3:** Write a 300-word opinion piece about AI music generation. Take a clear position. Support it with specific observations from your experiments today and from the ethical arguments presented above. This is for your journal, but write it as if you were publishing it.

Day 3: AI Mastering & Production Enhancement

What Mastering Is

Mastering is the final step in audio production — the process of taking a finished mix and optimizing it for distribution. A mastering engineer adjusts the overall EQ (frequency balance), dynamics (compression, limiting), stereo width, and loudness to ensure the track sounds polished and consistent across all playback systems (headphones, car speakers, phone speakers, club systems).

Professional mastering traditionally costs \$50-200+ per track with a human engineer. AI mastering tools now offer instant results for a fraction of the cost. For many purposes — demos, independent releases, audition recordings, portfolio tracks — AI mastering produces results that are genuinely good enough.

LANDR

LANDR (landr.com) is the most established AI mastering platform. Here is how to use it:

Step 1: Go to landr.com and create a free account. The free tier lets you preview mastered results but requires a subscription or per-track purchase to download full-quality files.

Step 2: Upload an audio file. This should be a mixed track — either a recording of your performance, a demo, or a project you have mixed. For best results, upload a WAV or AIFF file (uncompressed), though MP3 works.

Step 3: LANDR analyzes the audio and applies its AI mastering chain. You can choose from several intensity levels (low, medium, high) and styles (warm, balanced, open). Start with 'medium' intensity and 'balanced' style.

Step 4: Listen to the before-and-after comparison. Toggle between the original and mastered versions. Listen for: changes in clarity, changes in low-end response, changes in overall loudness, and whether any transients (attack moments) have been over-compressed.

Other AI Production Tools

iZotope Ozone — Professional-grade mastering software with an AI 'Master Assistant' that analyzes your track and suggests a starting point for your mastering chain. More control than LANDR but steeper learning curve.

BandLab Mastering — Free AI mastering built into BandLab's online music creation platform. Less sophisticated than LANDR but completely free.

CloudBounce — Another AI mastering service, often competitive with LANDR on quality. Worth trying on the same track and comparing results.

When to Use AI Mastering vs. a Human Engineer

Use AI mastering when:

- Creating demos or work-in-progress recordings
- Releasing music independently on a tight budget
- Preparing audition recordings that need to sound polished
- Quick turnaround is important
- The music will primarily be heard through casual listening (streaming, social media)

Use a human engineer when:

- Releasing a major project (album, EP) that represents your best work
- The music has complex dynamic range that needs careful handling (classical, jazz)
- You want creative mastering decisions, not just technical optimization
- The project will be heard in high-fidelity environments (concert halls, audiophile systems)

Exercises

♦ **Exercise 1:** If you have a recording of yourself playing (even a phone recording), upload it to LANDR and try the AI mastering. Compare the before and after. What improved? Did anything get worse?

♦ **Exercise 2:** If you do not have a recording ready, use BandLab (free) to record a short performance through your computer's microphone, then master it with BandLab's built-in AI mastering. The recording quality will not be studio-grade, but this teaches you the workflow.

♦ **Exercise 3:** Ask Claude: 'Explain the mastering process step by step — EQ, compression, limiting, stereo width, loudness — as if teaching a music student who has never mastered a track. For each step, explain what it does, why it matters, and what to listen for.' Use this as your reference guide when evaluating AI mastering results.

Day 4: AI Transcription, Notation & Lead Sheets

From Audio to Notation

One of the most time-consuming tasks in music is transcription — listening to a recording and writing out the notes. AI is getting remarkably good at this, though it still requires human review and correction, especially for complex music.

The Tools

Moises.ai (Chord Detection) — In addition to stem separation, Moises can detect chords in a recording and display them synced to the audio timeline. This is useful for quickly generating chord charts and lead sheets.

AnthemScore — Converts audio recordings to sheet music (MIDI and standard notation). Best for simple to moderately complex music. Works well for single-instrument recordings and chord-based music. Struggles with dense orchestral textures.

Basic Pitch (by Spotify) — A free, open-source audio-to-MIDI converter. It is simpler than AnthemScore but produces good results for monophonic (single-note) lines. Great for transcribing melodies and solos.

Noteflight / Flat.io — Online notation software with some AI-assisted features for transcription and arrangement.

Workflow: Creating a Lead Sheet from a Recording

Here is a practical workflow for creating a lead sheet (melody + chords) from a recording using currently available AI tools:

Step 1: Upload the track to Moises.ai. Get the chord detection and the stem separation.

Step 2: Isolate the melody instrument/voice stem. Listen to it clean.

Step 3: Use the chord detection to get the basic harmonic structure. Copy the chord symbols.

Step 4: Either transcribe the melody by ear (using the isolated stem makes this much easier) or run the stem through Basic Pitch to get a MIDI approximation.

Step 5: Import the MIDI into notation software (MuseScore, free) and clean it up manually. AI-generated MIDI is always approximate — you will need to adjust rhythms, fix enharmonic spellings, and correct notes that the AI got wrong.

Step 6: Add the chord symbols from Moises. Review the full lead sheet against the original recording.

The AI does not give you a perfect lead sheet, but it gives you a strong starting point that saves 60-80% of the time compared to transcribing entirely by ear.

Exercises

♦ **Exercise 1:** Use Moises.ai's chord detection on a song you know well. How accurate are the chords? Where does it get them right, and where does it make errors? Common errors include confusing relative major/minor, missing chord extensions (7ths, 9ths), and simplifying passing chords.

♦ **Exercise 2:** Try the lead sheet workflow described above with a simple song — something with a clear melody and standard chord changes. How much time does the AI save compared to transcribing entirely by ear?

♦ **Exercise 3:** Ask Claude to generate a chord analysis of a jazz standard you know: 'Analyze the chord changes to [standard name]. For each chord, identify its function (tonic, subdominant, dominant, secondary dominant, etc.) and any common substitutions.' Compare Claude's analysis to what you know about the tune.

Day 5: AI Voice Synthesis & Narration

What Voice Synthesis Is

AI voice synthesis creates realistic human-sounding speech from text. The technology has advanced to the point where it is genuinely difficult to distinguish AI-generated speech from real human speech. For musicians, this opens up creative and practical possibilities.

ElevenLabs

ElevenLabs (elevenlabs.io) is the leading AI voice synthesis platform. Here is how to use it:

Step 1: Go to elevenlabs.io and create a free account. The free tier gives you a limited number of characters per month — enough for several short projects.

Step 2: Type or paste the text you want spoken. This could be anything from a short introduction to a full narration.

Step 3: Choose a voice. ElevenLabs offers a library of pre-made voices with different genders, ages, accents, and speaking styles. Preview several to find one that fits your project.

Step 4: Adjust settings. You can control stability (how consistent the voice sounds) and clarity. Higher stability is better for formal narration; lower stability sounds more natural and expressive.

Step 5: Generate and download the audio.

Applications for Musicians

- **Narration for multimedia projects:** If you are creating a concept album, a musical theater piece, or a multimedia performance, AI narration can provide spoken elements without hiring a voice actor for the demo stage.
- **Educational content:** Create spoken lessons, guides, or explanations for your students or audience.
- **Podcast and video voiceover:** Generate professional-sounding narration for content about your music, your concerts, or your artistic process.
- **Accessibility:** Convert your written program notes or artist statements into audio for accessibility purposes.
- **Demo and prototype:** When pitching a multimedia project, AI narration lets you create a convincing demo of the spoken elements before investing in professional voice talent.

Important Ethical Note

Voice cloning — creating an AI voice that sounds like a specific real person — is possible with ElevenLabs and similar tools. Creating a clone of your own voice (for convenience or creative purposes) is generally acceptable. Cloning someone else's voice without their explicit permission is ethically problematic and potentially illegal. Never clone a voice you do not have rights to use.

Exercises

♦ **Exercise 1:** Go to ElevenLabs, create an account, and generate a spoken version of your artist bio. Try three different voices and compare them. Which one best represents how you want your professional brand to sound?

♦ **Exercise 2:** Generate a spoken introduction for a concert: 'Welcome to tonight's performance by [your name]...' Write a script that includes information about the program, then generate the audio. Could you use this at an actual performance?

♦ **Exercise 3:** Create a spoken version of one set of your program notes. This could be used as an audio companion for audience members who prefer listening, or as content for a podcast or social media.

Day 6: AI Audio Editing & Podcast Production

Descript: Editing Audio by Editing Text

Descript (descript.com) is an audio and video editing tool that uses AI to transcribe your recording into text, then lets you edit the audio by editing the text. Delete a sentence from the transcript, and Descript removes it from the audio. Rearrange paragraphs in the transcript, and the audio rearranges to match. This is revolutionary for anyone who needs to edit spoken audio.

Applications for musicians:

- Editing recordings of rehearsals, lessons, or masterclasses to extract key moments
- Creating podcasts about your music or your field
- Editing interviews for promotional use
- Removing 'ums,' 'uhs,' and filler words from recordings automatically
- Creating transcripts of spoken content for accessibility

Getting Started with Descript

Step 1: Go to descript.com and create a free account. The free tier includes transcription and basic editing.

Step 2: Upload an audio file. This could be a recording of yourself talking, a lesson, a rehearsal with discussion, or any spoken audio.

Step 3: Wait for the AI transcription. Descript will convert your audio to text, usually with high accuracy.

Step 4: Edit the text. Delete sections you do not want. Descript will remove the corresponding audio. Highlight filler words and remove them with one click.

Step 5: Export the edited audio.

Starting a Music Podcast

Many musicians are starting podcasts about their niche — instrument-specific conversations, music theory breakdowns, industry discussions, practice tips, or interviews with colleagues. AI tools make this remarkably accessible:

- Use Claude to brainstorm episode topics and write outlines
- Use ElevenLabs for intros and outros if you do not want to record your own
- Use Descript for editing (removing mistakes, tightening pacing)
- Use AI mastering tools to polish the audio quality
- Use Claude to write show notes and episode descriptions

Exercises

♦ **Exercise 1:** Record yourself talking for 3-5 minutes about a piece you are currently working on — what you find interesting about it, what challenges you face, what you want the audience to hear. Upload this to Descript. Edit the transcript to create a tight, polished 2-minute version. Export the edited audio.

♦ **Exercise 2:** Use Claude to outline a three-episode podcast series about a topic in your musical niche. Each episode should have a title, a hook, three main talking points, and a closing thought. This is your podcast blueprint — you could actually produce it using the tools you have learned.

Day 7: Building Your AI Music Production Workflow

Assembling Your Toolkit

Over the past six days, you have explored stem separation, music generation, mastering, transcription, voice synthesis, and audio editing. Now it is time to decide which tools earn a permanent place in your workflow and how they fit together.

Prompt: "I am a [your description] musician. This week I tried the following AI audio tools: Moises.ai for stem separation, Suno/Udio for music generation, LANDR for mastering, Moises/Basic Pitch for transcription, ElevenLabs for voice synthesis, and Descript for audio editing. Based on my specific musical activities — [list what you actually do: practice, perform, teach, record, compose, etc.] — which of these tools should be core parts of my workflow, and which are nice-to-know but not essential? Help me prioritize."

Document your personal AI audio toolkit in your journal. For each tool you plan to keep using, write down: what you will use it for, how often, and what the free tier covers versus what you would need to pay for.

Week 3 Reflection

In your journal, write about: How has your relationship to AI changed now that you have used tools that work directly with music — your core domain? What excited you most? What concerned you most? Has your ethical position from Week 1 evolved?

Week 4: AI Visual & Video Creation

Musicians are visual artists whether they realize it or not. Your album art, concert posters, social media graphics, music videos, and online presence all require visual content. Until recently, creating professional-quality visuals required either graphic design skills or a budget for hiring designers. AI has changed that equation entirely.

Day 1: AI Image Generation Fundamentals

How AI Image Generation Works

AI image generators create images from text descriptions (called 'prompts,' just like in text AI). The models were trained on millions of images paired with text descriptions, learning the visual patterns associated with words and concepts. When you describe an image, the AI generates pixels that match the patterns it learned.

The analogy for musicians: imagine a session musician who has played on thousands of recordings across every genre. When a producer says 'give me something funky in E-flat,' the musician draws on all that experience to generate something appropriate. AI image generators do the same thing with visual elements.

The Major Tools

ChatGPT with DALL-E — OpenAI's image generator, integrated directly into ChatGPT. Convenient because you can describe images in natural conversation. Produces clean, versatile images. Access through your ChatGPT account.

Midjourney — Often considered the best for artistic and aesthetic quality. Produces images with a distinctive, often painterly style. Originally operated through Discord (a messaging platform), now has a web interface at midjourney.com. Has a subscription model.

Canva AI (Magic Media) — Built into Canva, the popular graphic design platform. Less powerful than Midjourney or DALL-E as a standalone generator, but incredibly useful because the generated images integrate directly into Canva's design tools. Free tier available.

Adobe Firefly — Adobe's AI image generator, designed for commercial use with clear copyright policies. Integrates with Photoshop and other Adobe tools. Trained on licensed and public domain content, which matters for commercial use.

Ideogram — Excels at generating images that include text (logos, signs, posters). This is a notable strength — most AI image generators struggle with legible text in images.

How to Write Image Prompts

Image prompting is different from text prompting. Here are the key elements:

Subject — What is in the image. 'A cellist performing on stage.' 'A jazz club at midnight.' 'Hands on piano keys.'

Style — The artistic style. 'Oil painting,' 'minimalist vector illustration,' 'black and white photography,' 'watercolor,' 'retro poster art.'

Mood/Atmosphere — 'Warm golden lighting,' 'moody and dramatic,' 'bright and energetic,' 'intimate and quiet.'

Composition — How the image is framed. 'Close-up,' 'wide shot,' 'from above,' 'centered,' 'rule of thirds.'

Details — Specific visual elements. 'With string lights in the background,' 'wearing a red dress,' 'vintage microphone,' 'smoke and blue stage lighting.'

Technical qualities — 'High resolution,' '4K,' 'cinematic,' 'shallow depth of field,' 'film grain.'

Example prompt that uses all elements:

Prompt: "A jazz drummer in mid-performance, intense expression, sticks blurred with motion, close-up from a low angle, moody blue and amber stage lighting, smoke catching the spotlights, shot in the style of a classic Blue Note album cover photograph, high contrast black and white with warm toning, shallow depth of field, cinematic quality"

Exercises

♦ **Exercise 1:** Open ChatGPT and use DALL-E (type 'generate an image of...') to create five different images related to your musical life. Try: a concert poster concept, an album art concept, a profile photo background, a social media header, and an abstract image inspired by a piece you are performing. For each, iterate on the prompt at least twice to improve the result.

♦ **Exercise 2:** Create a free Canva account and try Canva's AI image generator (Magic Media). Generate an image and then use Canva's design tools to add text and create a complete social media graphic. This combined workflow — AI generation plus design tool — is extremely practical.

♦ **Exercise 3:** Try generating the same image concept with two different tools (e.g., ChatGPT and Canva, or Midjourney and DALL-E). Compare the results. Which tool produced something closer to your vision? Note the differences in style, quality, and 'feel.'

Day 2: Album Art, Posters & Promotional Graphics

Creating Album Art with AI

Album art is the visual identity of a musical project. It communicates genre, mood, and artistic intention before anyone presses play. AI can generate album art concepts that rival professional design — and even if you ultimately hire a designer, AI-generated concepts give you a clear visual vocabulary to communicate your vision.

Step-by-Step: Creating Album Art

Step 1: Define your vision. Before you prompt any AI tool, answer these questions:

- What is the mood of the music? (Dark, bright, ethereal, gritty, warm, cold)
- What genre signals do you want to convey? (Classical elegance, jazz cool, indie quirky, electronic sleek)
- What visual associations does the music evoke? (Landscapes, cityscapes, abstract shapes, human figures, nature)
- What color palette feels right? (Earth tones, neon, monochrome, pastel, high contrast)
- Are there any existing album covers you admire that capture a similar feeling?

Step 2: Use Claude to refine your vision into a prompt. This is a powerful technique — use one AI tool to write prompts for another.

Prompt: "I am creating album art for [describe the project — genre, mood, concept]. The music feels like [describe the emotional/visual associations]. I want the art to evoke [specific feelings]. I admire the visual style of [reference album covers or art styles]. Write me five detailed image generation prompts I can use in DALL-E or Midjourney to create album art concepts."

Step 3: Generate multiple concepts. Do not settle for the first image. Generate at least 10-15 variations across different prompts. Create a folder for all of them.

Step 4: Refine your favorite. Take the concept you like best and iterate: 'Same concept but warmer tones,' 'same concept but more contrast,' 'same concept but add [element].'

Step 5: Finalize in Canva. Import your favorite AI-generated image into Canva and add text (album title, artist name) using Canva's typography tools. This gives you a complete, polished album cover.

Concert Posters

Concert posters follow similar principles but with additional practical requirements: they need to communicate date, time, venue, performers, and ticket information clearly, while still being visually compelling.

Prompt: "Create a concert poster concept: [describe the visual style you want]. The design should have space for text overlays — a clear area at the top or bottom for the concert title, date, venue, and performer names. [Describe mood, color palette, and any specific visual elements]."

Tip: Most AI image generators struggle with text in images. Generate the visual concept with AI, then add all text in Canva, Photoshop, or a similar design tool. Never rely on AI to generate legible text in images (except Ideogram, which handles this better).

Exercises

♦ **Exercise 1:** Create album art for a real or hypothetical musical project. Go through all five steps: define your vision, generate prompts with Claude, create at least 10 concepts, refine your favorite, and add text in Canva. The final product should be something you would actually use.

♦ **Exercise 2:** Create a concert poster for your next performance (or a hypothetical one). Include all necessary information: event name, date, time, venue, performers. Use AI for the visual concept and Canva for layout and text.

Day 3: AI Video Generation for Musicians

The State of AI Video (2026)

AI video generation has made stunning progress. Tools can now generate short video clips (typically 5-30 seconds) from text prompts, and the quality ranges from 'interesting experiment' to 'genuinely cinematic.' For musicians, this opens up possibilities for music videos, visual accompaniments, social media content, and promotional material.

The Tools

Runway (runwayml.com) — The most established AI video generation tool. Generates short video clips from text prompts or from input images. The free tier gives you a limited number of generations. Strengths: cinematic quality, good camera movement, consistent style.

Pika (pika.art) — Generates video from text and images. Known for smooth, natural-looking motion. Also has features for extending existing videos and adding effects.

Kling — An AI video tool that excels at longer clips and realistic human motion. Produces some of the most photorealistic AI video currently available.

Sora (by OpenAI) — When available, produces some of the highest-quality AI video. Access may be limited.

Practical Applications for Musicians

Visualizers and Lyric Videos

Generate abstract visual content that accompanies your music on YouTube or social media. This is one of the most immediately practical applications: instead of posting audio-only content (which performs poorly on most social platforms), you can pair your music with compelling visuals.

Promotional Clips

Create short, visually striking clips for social media that promote upcoming events. A 5-10 second AI-generated visual with text overlay can be more eye-catching than a static image.

Music Video Concepts

Use AI video to prototype music video concepts before committing to a full production. Generate short scenes to show a director or collaborator: 'This is the visual mood I am imagining for this section of the piece.'

Exercises

♦ **Exercise 1:** Go to runwayml.com, create a free account, and generate three different short video clips from text descriptions. Try: an abstract visualization that matches the mood of a piece you are performing, a cinematic scene that evokes the emotion of your music, and a promotional-style clip for an upcoming event.

♦ **Exercise 2:** Create a 'visual mood board' for a hypothetical music video. Use Claude to brainstorm the concept, then generate 5-8 AI images that represent different scenes or moods from the video. Arrange them in sequence. This is exactly how a real director storyboards a music video — and now you can do it with AI in an afternoon.

Day 4: Music Videos & Visual Content with AI

Today you put the pieces together. Using a combination of AI image generation, AI video generation, and free editing tools, you will create a complete visual piece that accompanies your music.

Creating a Visualizer

Here is the complete workflow:

Step 1: Choose a piece of your music (a recording, even a phone recording).

Step 2: Use Claude to describe the visual mood: 'My piece starts contemplative, builds to a climax at 2:30, then resolves peacefully. Describe what a visual accompaniment might look like at each stage.'

Step 3: Generate 5-8 images that represent different moments in the piece.

Step 4: Generate 2-3 short video clips for the most dynamic moments.

Step 5: Use a free video editor (CapCut, DaVinci Resolve, or even iMovie) to combine the images and clips into a sequence. Set them to your music. Add transitions.

Step 6: Export and post to YouTube or social media.

Exercises

♦ **Exercise:** Complete the visualizer workflow above with a piece of your own music. The result does not need to be perfect — it is a learning exercise. But push yourself to create something you would actually share on social media.

Day 5: AI Presentations & Pitch Decks

Musicians pitch constantly — to venues, to funders, to residency programs, to school administrators. AI can create compelling slide presentations that look professional.

Tools for AI Presentations

Canva Presentations — Canva has AI-powered presentation features, including Magic Design (generates slide layouts) and AI-generated images. The free tier is very capable.

Gamma (gamma.app) — An AI-first presentation tool. Describe your presentation, and Gamma generates a complete slide deck with layout, content, and visuals. Excellent for quick, polished results.

Beautiful.ai — AI-powered presentation design that automatically adjusts layout as you add content.

Workflow: Creating a Pitch Deck

Step 1: Use Claude to outline your pitch.

Prompt: "I need to create a pitch deck for [describe what you are pitching] to [describe your audience]. The deck should be [number] slides. Outline the content for each slide: the headline, the key message, and any supporting data or details."

Step 2: Take Claude's outline to Gamma or Canva. Input the content and let the AI generate the visual design.

Step 3: Customize the design to match your brand. Replace any AI-generated images that do not fit.

Step 4: Review and rehearse. AI makes the deck — you make the pitch.

Exercises

♦ **Exercise:** Create a pitch deck for a project you care about — a concert series proposal, a grant presentation, an educational program pitch, or a collaboration proposal. Use Claude for content and Gamma or Canva for design. The deck should be 8-12 slides.

Day 6: Branding & Visual Identity with AI

Your visual identity — logo, colors, fonts, and overall aesthetic — should be consistent across everything: your website, your social media, your posters, your business cards, and your EPK. AI can help you develop a cohesive brand identity.

Developing Your Brand with AI

Prompt: "I am a [your description] musician building a visual brand identity. My music is [describe the aesthetic/mood]. My target audience is [describe]. I admire the visual branding of [name 2-3 musicians or artists whose branding you like]."

Help me develop: (1) a color palette — 3-5 colors with hex codes that capture my artistic identity, (2) font recommendations — a heading font and a body font that pair well and match my brand, (3) a description of my overall visual style in 2-3 sentences that I can reference whenever creating visual content, (4) suggestions for visual motifs or recurring elements that could become part of my brand."

Once you have your brand guidelines (even informal ones), use them consistently across everything you create. Tell Claude and your image generators about your brand: 'My color palette is [colors], my aesthetic is [description]. Generate an image that fits my brand.'

Exercises

♦ **Exercise:** Develop your brand guidelines using the prompt above. Then create three pieces of visual content (a social media post, a concert promotion graphic, and a profile header) that all follow your brand. They should look like they belong together.

Day 7: Review & Visual Portfolio

Assemble everything you created this week into a visual portfolio. You should have:

- Multiple AI-generated images in different styles
- At least one album art concept
- At least one concert poster

- Short AI-generated video clips
- A visualizer or visual companion for a piece of your music
- A pitch deck
- Brand guidelines

Create a folder called 'Visual Portfolio' and organize these assets. Note which tools and prompts produced your best results — you will reuse these approaches throughout your career.

Week 5: AI Automation, Workflows & Business Tools

This week shifts from creating content to creating systems. You are going to learn how to automate the repetitive tasks that eat into your practice and creative time, build professional tools like an Electronic Press Kit, and discover how AI can transform your teaching and performance preparation.

Day 1: No-Code AI Platforms & Why They Matter

What No-Code Means

'No-code' tools let you build functional digital products — websites, apps, automations, databases — without writing any code. You work with visual interfaces: dragging and dropping elements, connecting blocks, filling in fields. In 2026, no-code tools powered by AI are extraordinarily capable. You can build things that would have required a developer five years ago.

Key No-Code Platforms for Musicians

Notion — An all-in-one workspace for notes, databases, project management, and collaboration. Notion AI adds AI writing and analysis directly into your workspace. Free for personal use.

Airtable — A spreadsheet-database hybrid that is perfect for tracking gigs, contacts, repertoire, and finances. More powerful than a spreadsheet, easier than a database.

Zapier — Connects different apps together. For example: 'When I get a new email with the subject 'Booking Inquiry,' automatically create an entry in my Airtable gig tracker and send me a Slack notification.' Free tier allows 5 automations.

Make (formerly Integromat) — Similar to Zapier but more powerful for complex automations. Steeper learning curve but more flexibility.

Exercises

♦ **Exercise 1:** Create a free Notion account and set up a workspace for your music career. Create pages for: Repertoire List, Gig History, Contact Directory, and Practice Log. Use Notion AI to help you structure each page.

♦ **Exercise 2:** Create a free Airtable account and build a simple 'Gig Tracker' database with fields for: Date, Venue, Event Name, Pay, Contact Person, Status (confirmed/pending/completed), and Notes.

♦ **Exercise 3:** Create a free Zapier account and set up one simple automation. Example: 'When I receive an email with the word 'performance' in the subject, automatically forward it to my music-specific email folder.' Even simple automations teach you the logic of how these tools work.

Day 2: Connecting AI Tools Together

The Power of Integration

Individual AI tools are useful. AI tools connected together are transformative. Today you learn how to create automated workflows that chain multiple tools together.

Workflow Examples for Musicians

The Concert Announcement Pipeline

Trigger: You add a new confirmed gig to your Airtable gig tracker.

Step 1: Zapier detects the new entry.

Step 2: Zapier sends the gig details to Claude (via API or Claude's Zapier integration) with a prompt: 'Write a concert announcement social media post based on these details.'

Step 3: The generated post is saved to a Google Doc for your review.

Step 4: After you review and approve, it gets scheduled in your social media tool.

The Student Management Pipeline

If you teach, you can automate administrative tasks:

When a student books a lesson (via Calendly or Google Calendar), automatically: create a lesson record in Airtable, send a confirmation email with practice assignment from the previous lesson, and add the student's repertoire to your tracking system.

Exercises

♦ **Exercise 1:** Design (on paper) an automated workflow for something in your music life. Map out: What triggers it? What steps happen automatically? What still needs your human input? You do not need to build it today — just design it clearly.

♦ **Exercise 2:** Build one simple multi-step automation in Zapier. Start small: two or three connected steps. Test it and make sure it works.

Day 3: AI for Scheduling, Email & Admin

The administrative side of a music career — scheduling, email, invoicing, record-keeping — eats an enormous amount of time. AI can help with all of it.

AI Email Management

Use Claude to batch-process email tasks:

Prompt: "I have 15 emails I need to respond to. I am going to paste each one. For each, draft a response that is professional, concise, and matches my communication style [describe your style — casual/formal, warm/direct, etc.]. I will review and edit each response before sending."

AI Scheduling

Tools like Calendly, Cal.com (free), and Google Calendar's scheduling features automate the back-and-forth of scheduling meetings, lessons, and rehearsals. Claude can help you set these up:

Prompt: "Help me set up a scheduling system for my private teaching studio. I teach on [days] from [times]. Each lesson is [length]. I need: a booking page students can access, automatic reminders, and a way to track cancellations and makeup lessons. What is the simplest free tool to accomplish this, and walk me through setting it up step by step."

Exercises

♦ **Exercise 1:** Use Claude to draft responses to five real emails in your inbox. Review and send them. Track how much time this saves compared to writing each one from scratch.

♦ **Exercise 2:** Set up a scheduling tool (Cal.com is free) for one recurring appointment type — lessons, office hours, or rehearsals.

Day 4: Building an AI-Powered Electronic Press Kit (EPK)

What an EPK Is

An Electronic Press Kit is a one-stop collection of everything a presenter, venue, festival, or media outlet needs to know about you: bio, photos, recordings, videos, press quotes, technical requirements, and contact information. Every professional musician needs one, and AI can help you build an excellent one in a single afternoon.

Building Your EPK Step by Step

Step 1: Gather your assets.

- Professional bio (you wrote this in Week 2)
- Professional photos (even good smartphone photos work to start)
- Recording samples or links (SoundCloud, YouTube, Spotify)
- Video clips of live performances
- Press quotes or reviews (if you have them)
- Technical rider / stage requirements (if applicable)
- Contact information

Step 2: Use Claude to help write any missing elements.

Prompt: "I need to create an Electronic Press Kit. I have my bio and recordings but I'm missing: a compelling one-paragraph summary for the EPK landing page, a list of notable performances formatted for press use, and a brief quote about my artistic vision that could serve as a pull quote. Here is my bio for context: [paste bio]. Help me create each of these missing elements."

Step 3: Build the EPK using a free platform.

- Canva: Create a beautiful PDF EPK using their presentation or document templates
- Notion: Create a clean web-based EPK that you can share as a link
- Google Sites: Build a simple EPK website (free, easy, no code)
- Your website: Dedicate a page of your website (which you will build in Week 6) to your EPK

Exercises

♦ **Exercise:** Build your EPK today. Gather your assets, use Claude to fill in any gaps, and assemble everything in Canva, Notion, or Google Sites. By the end of the day, you should have a shareable EPK that you could send to a venue tomorrow.

Day 5: AI for Teaching & Pedagogy

AI as a Teaching Assistant

If you teach music (or plan to), AI is the most powerful pedagogical tool to emerge in your lifetime. It does not replace you as a teacher — your musical ear, your ability to demonstrate, your understanding of each student's unique needs remain irreplaceable. But AI can handle many of the time-consuming tasks that surround teaching, freeing you to focus on what matters most: the music and the student.

Lesson Planning

Prompt: "I teach [instrument] to a [age/level] student who is currently working on [specific repertoire] and struggling with [specific challenge]. Create a detailed 45-minute lesson plan that addresses the challenge"

through: (1) a warm-up exercise targeting the specific technical issue, (2) an etude or study that develops the skill in isolation, (3) guided practice on the repertoire passage, and (4) a creative exercise that makes the skill fun. Include time allocations for each section."

Curriculum Design

Prompt: "Help me design a one-semester curriculum for [course description]. The students are [level/background]. The course meets [frequency] for [duration]. I want to cover [topics]. Create a week-by-week syllabus with: weekly topics, learning objectives, assignments, and suggested repertoire/materials."

Assessment and Feedback

Prompt: "I need to write semester evaluations for my students. For each student, I will give you their name, level, repertoire covered, strengths, and areas for improvement. Write a personalized, encouraging but honest evaluation paragraph for each one. Tone: warm and professional, like a teacher who genuinely cares about each student's development."

Creating Practice Materials

Prompt: "Create a set of 10 progressive exercises for [specific technique] on [instrument], going from beginner to intermediate level. For each exercise, include: the notation or description, the purpose, common mistakes to watch for, and how the student knows they have mastered it."

Exercises

♦ **Exercise 1:** Create a detailed lesson plan for a student you currently teach (or a hypothetical student). Use the lesson planning prompt template. Evaluate the result: would you actually use this plan?

♦ **Exercise 2:** Design a four-week mini-curriculum for a group class, masterclass series, or workshop you could teach. Use AI to develop the week-by-week structure, then add your own expertise to refine it.

Day 6: AI for Live Performance & Shows

Pre-Performance Preparation

Prompt: "I have a concert in [timeframe]. I am performing [repertoire]. Help me create a performance preparation checklist that covers: (1) musical preparation milestones (two weeks out, one week out, day before, day of), (2) logistical checklist (what to bring, what to confirm with the venue, what to wear), (3) mental preparation strategies for managing performance anxiety, and (4) a post-concert routine for recovery and reflection."

Stage Announcements and Spoken Introductions

Prompt: "Write spoken introductions for each piece on my concert program. The audience is [describe]. Each introduction should be 30-60 seconds when spoken — informative but conversational, as if I am talking to the audience like friends. My program: [list pieces]. I want to connect the pieces thematically, so include transitions between the introductions."

Setlist Planning

Prompt: "Help me plan the setlist order for my concert. Here are the pieces I am performing: [list with approximate durations and key information about each]. I want the program to have a dramatic arc — a compelling opening, varied pacing, a strong closer, and logical flow between pieces. Suggest an order and explain your reasoning for the sequencing."

Exercises

♦ **Exercise:** Create a complete performance preparation package for your next concert or recital: checklist, spoken introductions for each piece, and a setlist order rationale. Even if your program is already set, going through this exercise strengthens your preparation.

Day 7: Assembling Your Personal AI Toolkit

Over five weeks, you have explored dozens of AI tools. Today you formalize your personal toolkit — the specific tools and workflows you will use going forward.

Prompt: "Based on my musical activities — I am a [description] who [list what you do: perform, teach, compose, record, produce, manage, etc.] — help me create my personal AI toolkit. For each of my activities,

recommend: the specific AI tool to use, how to use it (brief workflow), and whether the free tier is sufficient or I should budget for a paid plan. Be ruthlessly practical — only recommend tools I will actually use regularly."

Document your toolkit in your journal. Include the tool name, what you use it for, your login information (stored securely), and any custom prompt templates you have developed for that tool.

Week 6: Building Your Website with Claude Code

This week you build and launch your professional musician website — and you are going to do it with full control. Instead of using a website builder that limits your options and charges you monthly, you are going to use Claude Code, an AI development partner that runs in your terminal, to create a custom website that you own completely. Then you will deploy it for free on Vercel or Cloudflare Pages, platforms used by professional developers worldwide. The result: a fast, beautiful, professional site that costs you nothing to host and that you can change anytime you want.

Day 1: Why Every Musician Needs a Website (and Full Control of It)

Why a Website

Social media platforms come and go (remember Myspace?), but your website is your permanent home on the internet. It is the one digital space you fully control — your design, your content, your branding. When a presenter Googles your name, your website should be the first result.

Your website needs to accomplish five things:

- Establish credibility: A professional website signals that you take your career seriously
- Provide information: Who you are, what you do, how to book/contact you
- Showcase your work: Recordings, videos, photos, programs
- Convert visitors: Give people a clear next step — book you, attend a concert, join your email list
- Be findable: Search engine optimization (SEO) so people can discover you

Why Full Control Matters

Website builders like Wix and Squarespace are convenient, but they come with tradeoffs: monthly fees (\$16-30/month adds up over a career), limited customization, your content locked inside their platform, and your site disappears if you stop paying. Building your own site gives you:

- Zero hosting cost — Vercel and Cloudflare Pages host static sites for free, permanently
- Complete design freedom — every pixel is yours to control
- True ownership — your code, your files, your domain, no lock-in
- Professional credibility — a custom site signals technical sophistication

- A foundation for growth — once you have a site, you can add features (a blog, a booking system, a store) whenever you want

And here is the key: you do not need to learn to code to do this. Claude Code writes the code for you. You describe what you want in plain English, and it builds it. Your job is to direct the creative vision and review the results.

What Your Website Needs

At minimum, your musician website needs these pages:

- Home: Your name, what you do, a compelling tagline, and a professional photo. This page should answer 'who is this person and why should I care?' in five seconds.
- About / Bio: Your full bio and artist statement
- Media: Recordings, videos, and photos (embedded from YouTube, SoundCloud, Spotify, etc.)
- Calendar / Events: Upcoming performances
- Press / EPK: Your electronic press kit (built in Week 5)
- Contact: How to reach you, with a contact form
- Teaching (if applicable): Your teaching studio information, rates, and how to sign up

Exercises

♦ **Exercise:** Before touching any tool, use Claude (at claude.ai) to plan your site structure. Prompt: 'I am building a personal musician website. Help me plan the page structure, what content goes on each page, and the overall user flow — what should a visitor see first, and what action should they take? Also suggest a visual style that fits my brand: [describe your aesthetic from Week 4].' Review and refine until you are happy with the plan. Save this — you will give it to Claude Code tomorrow.

Day 2: The Terminal & Setting Up Your Tools

What the Terminal Is

The terminal (also called the command line, shell, or console) is a text-based way to talk to your computer. Instead of clicking icons and dragging windows, you type commands. It sounds old-fashioned, but it is how every developer works — and it is how Claude Code communicates with your computer. You need to be comfortable enough with the terminal to open it, navigate to folders, and run commands.

Opening the Terminal

On Mac: Open Finder, go to Applications → Utilities → Terminal. Or press Command+Space, type 'Terminal,' and press Enter. You will see a window with a text prompt, probably ending in \$ or %. This is where you type commands.

On Windows: Press the Windows key, type 'PowerShell,' and open Windows PowerShell. Alternatively, install Windows Terminal from the Microsoft Store for a better experience.

Essential Terminal Commands

You only need a handful of commands to work with Claude Code. Here they are, with exactly what they do:

pwd — (Print Working Directory) Shows you where you currently are in your computer's file system. Type 'pwd' and press Enter. You will see something like '/Users/yourname' — that is your home folder.

ls — (List) Shows you all the files and folders in your current location. Type 'ls' and press Enter. You will see a list of names.

cd foldername — (Change Directory) Moves you into a folder. Type 'cd Documents' to move into your Documents folder. Type 'cd ..' to go back up one level. Type 'cd ~' to go back to your home folder.

mkdir foldername — (Make Directory) Creates a new folder. Type 'mkdir my-website' to create a folder called 'my-website.'

clear — Clears the screen if it gets cluttered. Just type 'clear' and press Enter.

Setting Up Your Tools

You need to install three things. Follow these steps in order.

Tool 1: Node.js. Go to nodejs.org in your browser. Download the LTS (Long Term Support) version — this is the stable, recommended version. Run the installer and follow the prompts. When it finishes, verify the installation by opening your terminal and typing:

```
node --version
```

You should see a version number like 'v22.x.x'. If you see 'command not found,' the installation did not work — close and reopen your terminal, then try again. If it still fails, ask Claude at claude.ai for help troubleshooting.

Tool 2: A code editor (Visual Studio Code). Go to code.visualstudio.com and download the free version for your operating system. Install it. This is where you will view and occasionally edit the files that Claude Code creates. You do not need to know how to use it deeply — it is mostly for viewing.

Tool 3: Git. Git tracks changes to your code over time. It is essential for deployment (Vercel and Cloudflare pull your code from Git). Install it:

Mac: Open Terminal and type 'git --version'. If Git is not installed, your Mac will prompt you to install the Xcode Command Line Tools. Click Install and wait.

Windows: Download from git-scm.com. Run the installer with default settings.

After installing, configure Git with your name and email (these appear in your code history):

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

Exercises

♦ **Exercise 1:** Open your terminal. Type 'pwd' to see where you are. Type 'ls' to see what is there. Use 'cd' to navigate to your Documents folder. Create a new folder called 'my-website' using 'mkdir my-website'. Navigate into it with 'cd my-website'. Type 'pwd' to confirm you are inside it. These five commands are all you need.

♦ **Exercise 2:** Verify all three tools are installed. In your terminal, type each of these and confirm you see version numbers: 'node --version', 'git --version', and 'code --version' (this last one verifies VS Code's command-line tool is available).

Day 3: Meet Claude Code — Your AI Development Partner

What Claude Code Is

Claude Code is a version of Claude that runs directly in your terminal. Instead of chatting in a browser window, you have a conversation with Claude right where your code lives. The key difference: Claude Code can see your files, create new files, edit existing files, and run commands on your computer. It is like having an expert developer sitting next to you, looking at the same screen, who can type faster than any human.

You tell Claude Code what you want in plain English. It writes the code, creates the files, and shows you the result. If something does not look right, you tell it what to change. This is not a shortcut or a cheat — it is the future of how software gets built. Professional developers use AI coding assistants daily.

Installing Claude Code

Open your terminal and type this command:

```
npm install -g @anthropic-ai/claude-code
```

This downloads and installs Claude Code globally on your computer. The command may take a minute to finish. When it is done, verify the installation by typing:

```
claude --version
```

You should see a version number. If you see an error, try closing and reopening your terminal.

Setting Up Your Account

Claude Code needs to connect to your Anthropic account. The first time you run it, type:

```
claude
```

It will open a browser window asking you to sign in to your Anthropic account (the same account you use at claude.ai). Sign in and authorize the connection. Claude Code usage is billed through your Anthropic account — check the current pricing and plans at claude.ai/code before starting, as pricing may have changed since this handbook was written.

Your First Claude Code Session

Navigate to your my-website folder in the terminal:

```
cd ~/Documents/my-website
```

Now start Claude Code:

```
claude
```

You are now in a conversation with Claude Code. It can see that you are in the my-website folder. Try typing:

Prompt: "Create a simple HTML file called index.html with a heading that says 'Hello, I am [your name]' and a paragraph that says 'This is my musician website.' Make it look clean and modern with some basic CSS styling."

Watch what happens. Claude Code will create the file, write the HTML and CSS, and tell you what it did. You can open the file in your browser to see the result: on Mac, type 'open index.html' in a new terminal tab; on Windows, type 'start index.html'.

This is the core workflow: you describe what you want, Claude Code builds it, you review the result, and you refine. That is it. That is all of software development, simplified to a conversation.

Key Claude Code Commands

claude — Starts a new Claude Code session in the current folder.

/clear — Clears the conversation history if things get confused.

Ctrl+C — Exits Claude Code and returns to the normal terminal.

Describing changes — Just tell Claude Code what you want in English: 'Make the heading bigger,' 'Add a navigation menu,' 'Change the background color to dark blue.' It reads your files, makes the changes, and explains what it did.

Exercises

♦ **Exercise 1:** Install Claude Code, set up your account, and have your first session. Create that simple index.html page and open it in your browser. Make three changes by asking Claude Code in plain English — change a color, add a paragraph, resize the heading. Watch how it edits the file each time.

♦ **Exercise 2:** Ask Claude Code: 'Explain what this index.html file contains, line by line, as if I am someone who has never seen HTML before.' Read its explanation. You do not need to memorize HTML, but understanding the basics — tags, elements, how structure and style relate — helps you direct Claude Code more effectively.

Day 4: Building Your Site with Claude Code

The Build

Today is the big day. You are going to build your full musician website in a single session. Set aside extra time if you can — 90 minutes instead of 60 is ideal.

Navigate to your my-website folder and start Claude Code. Then give it a comprehensive description of what you want. The more detail you provide, the better the result. Here is a template you should customize thoroughly:

Prompt: "Build me a complete, professional musician website. Here are the details:

MY INFORMATION:

Name: [your name]

Instrument/focus: [your instrument or area]

Tagline: [a short phrase, e.g., 'Violinist. Educator. Collaborator.']

PAGES NEEDED:

- 1. Home page — hero section with my name, tagline, and a professional photo placeholder. Clean, elegant design.*
- 2. About page — space for my bio (I will paste it in later) and artist statement.*
- 3. Media page — embedded audio player or links to SoundCloud/Spotify, embedded YouTube videos, and a photo gallery.*
- 4. Events page — a list of upcoming performances with dates, venues, and links.*
- 5. Contact page — a contact form (name, email, message) and my professional email.*

DESIGN REQUIREMENTS:

Color palette: [your brand colors from Week 4, e.g., deep navy, warm gold, cream]

*Style: [e.g., clean and modern, warm and elegant, bold and dramatic]
Fonts: [e.g., serif headings, clean sans-serif body text]
The site must look good on both desktop and mobile (responsive design).
Include a navigation menu that works on all screen sizes.*

TECHNICAL REQUIREMENTS:

Use plain HTML, CSS, and JavaScript — no frameworks. Keep it simple and fast-loading.

Create a clean folder structure with separate files for each page.

Include proper meta tags for SEO (title, description, Open Graph tags)."

Claude Code will create multiple files — HTML pages, a CSS stylesheet, possibly some JavaScript for the navigation menu and contact form. Let it work. When it finishes, open the index.html file in your browser to see the result.

Refining Your Site

The first version will not be perfect, and that is expected. Now you iterate. Look at each page in your browser and tell Claude Code what to change:

- 'The heading font is too small. Make it bigger and bolder.'
- 'I want the navigation menu to be sticky — it should stay at the top when I scroll.'
- 'The events page needs a cleaner layout. Put each event in a card with the date on the left and details on the right.'
- 'Add a footer to every page with my copyright, social media links, and a link to my email list.'
- 'The mobile version looks cramped. Add more spacing between sections.'
- 'Change the hero section background to a dark gradient and make the text white.'

Keep iterating until you are happy. There is no limit — Claude Code can make as many changes as you need. This is your site, and it should look exactly how you envision it.

Adding Your Real Content

Once the design is right, add your actual content. You can either edit the HTML files directly in VS Code (find the placeholder text and replace it with your real text) or tell Claude Code:

Prompt: "Replace the placeholder bio text on the About page with this: [paste your bio from Week 2]"

Prompt: "Add these upcoming events to the Events page: [list your real events with dates and venues]"

Prompt: "Embed this YouTube video on the Media page: [paste YouTube URL]"

Exercises

♦ **Exercise:** Build your complete website with Claude Code. Iterate until every page looks professional and matches your brand. Add your real content — bio, events, media links. By the end of today, you should have a complete website on your computer that you are proud of.

Day 5: Deploying to Vercel or Cloudflare Pages

What Deployment Means

Right now your website only exists on your computer. Deployment means putting it on a server — a computer that is always on, always connected to the internet — so that anyone in the world can visit it by typing a URL. The platforms we will use (Vercel and Cloudflare Pages) host static websites for free, permanently. No monthly fees, no ads, no catch.

Step 1: Create a GitHub Account and Repository

Vercel and Cloudflare deploy your site by pulling the code from GitHub — a platform that stores your code in the cloud. You need a GitHub account.

Go to github.com and create a free account if you do not have one. Then create a new repository (a project container) for your website:

- Click the '+' icon in the top right corner and select 'New repository'
- Name it something like 'my-music-website' (no spaces, use hyphens)
- Leave it set to 'Public' (free hosting requires public repositories on the free tier)
- Do NOT check any of the initialization options (no README, no .gitignore)
- Click 'Create repository'

GitHub will show you a page with setup instructions. Keep this page open — you will need the URL it shows (something like 'https://github.com/yourusername/my-music-website.git').

Step 2: Push Your Code to GitHub

In your terminal, navigate to your my-website folder and run these commands. You can ask Claude Code to help you, or type them directly:

```
git init
```

```
git add .
```

```
git commit -m "Initial website"
```

```
git remote add origin https://github.com/YOURUSERNAME/my-music-website.git
```

```
git branch -M main
```

```
git push -u origin main
```

Here is what each command does: 'git init' tells Git to start tracking this folder. 'git add .' stages all your files to be saved. 'git commit' saves a snapshot of your code with a message describing it. The 'git remote' command tells Git where your GitHub repository lives. 'git push' uploads your code to GitHub.

If you see an authentication prompt, GitHub may ask you to create a Personal Access Token instead of using your password. Follow the prompts or ask Claude Code: 'Help me authenticate with GitHub from the terminal.'

Step 3: Deploy on Vercel (Option A)

Vercel (vercel.com) is one of the most popular deployment platforms. It is used by companies like The Washington Post and Notion. Here is how to deploy your site:

1. Go to vercel.com and click 'Sign Up.' Sign up with your GitHub account (this connects them).
2. Click 'Add New' → 'Project.'
3. You will see your GitHub repositories listed. Click 'Import' next to your music website repository.
4. Vercel auto-detects that this is a static site (plain HTML/CSS/JS). You do not need to change any settings.
5. Click 'Deploy.'
6. Wait 30-60 seconds. Vercel builds and deploys your site.
7. When it finishes, Vercel gives you a URL like 'my-music-website.vercel.app.' Click it. Your website is now live on the internet.

From now on, every time you push changes to GitHub (using 'git add .', 'git commit', 'git push'), Vercel automatically redeploys your site within seconds. This is called continuous deployment — you push code, the live site updates. It is effortless.

Step 3: Deploy on Cloudflare Pages (Option B)

Cloudflare Pages is an excellent alternative to Vercel. It is completely free, extremely fast (Cloudflare runs one of the world's largest content delivery networks), and reliable.

1. Go to pages.cloudflare.com and create a free Cloudflare account (or sign in if you have one).
2. In your Cloudflare dashboard, go to 'Workers & Pages' in the sidebar.
3. Click 'Create' → 'Pages' → 'Connect to Git.'
4. Authorize Cloudflare to access your GitHub account and select your website repository.

5. In the build settings, leave the build command empty and set the output directory to '/' (since your HTML files are in the root of your project).

6. Click 'Save and Deploy.'

7. Wait 1-2 minutes. Cloudflare gives you a URL like 'my-music-website.pages.dev.' Your site is live.

Like Vercel, Cloudflare Pages automatically redeploys when you push to GitHub.

Which Should You Choose?

Both are excellent. Vercel is slightly simpler for first-time users. Cloudflare Pages is slightly faster globally. You cannot go wrong with either. Pick one and proceed.

Exercises

♦ **Exercise 1:** Push your website code to GitHub using the steps above. Verify you can see your files on github.com in your repository.

♦ **Exercise 2:** Deploy your site on Vercel or Cloudflare Pages. When you get your live URL, open it on your phone as well as your computer. Your site is now live on the internet.

♦ **Exercise 3:** Make a small change to your site using Claude Code (change a heading, update a color). Then push the change to GitHub: 'git add .', 'git commit -m "Updated heading"', 'git push'. Watch as your live site updates automatically within a minute.

Day 6: Custom Domains, SEO & Email Lists

Adding a Custom Domain

Your site is live at something like 'my-music-website.vercel.app.' That works, but for a professional presence, you want your own domain — like 'yourname.com' or 'yournamemusic.com.'

Step 1: Buy a domain. Go to a domain registrar — Namecheap (namecheap.com), Google Domains, or Cloudflare Registrar (if you used Cloudflare Pages, this is the most seamless option). Search for your desired domain. Most .com domains cost \$8-15 per year. This is the only cost associated with your website.

Step 2: Connect it to your deployment platform. The process varies by platform, but the general flow is: go to your project settings on Vercel or Cloudflare, click 'Domains' or 'Custom Domains,' type in the domain you bought, and follow the instructions to update your domain's DNS settings. If this feels confusing, ask Claude Code: 'Walk me through connecting my domain [yourdomain.com] from [registrar] to [Vercel/Cloudflare Pages].' It will give you step-by-step instructions specific to your setup.

Step 3: Wait for propagation. After updating DNS settings, it can take up to 24 hours (usually much less) for the change to take effect worldwide. Both Vercel and Cloudflare automatically set up HTTPS (the secure padlock in the browser) for your custom domain — you do not need to do anything extra.

SEO Basics

SEO (Search Engine Optimization) determines whether people find you when they search Google. Ask Claude Code to help:

Prompt: "Review my website files and optimize them for SEO. I want to rank for searches like '[your instrument] player [your city]' and '[your name] musician.' Update the page titles, meta descriptions, heading structure, alt text for images, and Open Graph tags. Also create a sitemap.xml file."

Setting Up an Email List

An email list is the most valuable digital asset a musician can own. Unlike social media followers (who are at the mercy of algorithm changes), your email list is yours. Use Mailchimp (free for up to 500 subscribers) or Buttondown (free for up to 100).

Sign up for your chosen email service, create a list, and get the embed code for a sign-up form. Then tell Claude Code:

Prompt: "Add an email newsletter sign-up form to my website. Here is the embed code from Mailchimp: [paste code]. Place it in the footer of every page and as a prominent call-to-action on the home page."

Use Claude at claude.ai to write your first newsletter:

Prompt: "Write a welcome email for new subscribers to my musician mailing list. It should: thank them for subscribing, tell them what to expect (how often I will email, what kind of content), share something personal or interesting to make a good first impression, and include a link to [a recording, video, or piece of content]. Tone: warm, personal, not corporate."

Exercises

♦ **Exercise 1:** If you are ready to invest \$8-15/year, buy a domain and connect it to your site. If not, the free `.vercel.app` or `.pages.dev` URL works fine for now.

♦ **Exercise 2:** Have Claude Code optimize your site for SEO. Push the changes to GitHub and verify your live site updates.

♦ **Exercise 3:** Set up an email list service and add a sign-up form to your website.

Day 7: Launch Day

Your website is live on the internet. Today you announce it and get feedback.

Use Claude at claude.ai to write announcement posts for each of your social media platforms.

Send an email to your contacts sharing the new site.

Ask five trusted colleagues or friends to review the site and give feedback.

Update your email signature, social media bios, and any online profiles with the new URL.

If any feedback reveals issues — a page that looks wrong on mobile, a broken link, a typo — fix them with Claude Code and push to GitHub. Your site updates automatically.

Week 6 Reflection

You just built and deployed a professional website using a terminal, an AI development partner, and professional-grade hosting — tools that most musicians have never touched. The skills you used this week (terminal navigation, working with Claude Code, pushing to GitHub, deploying to a server) are the foundation for the application development you will do in Weeks 8-10. You are no longer just an AI user — you are an AI builder.

Week 7: AI for Your Music Career & Business

Music is art, but a music career is a business. This week covers the business side — grants, finances, contracts, networking, and tour planning — with AI as your partner.

Day 1: AI for Grant Writing & Funding Applications

Why Grants Matter

Grants are free money for your art. Unlike loans, you do not pay them back. Unlike gig pay, you receive them before doing the work. The catch: grant applications are competitive, time-consuming, and require strong writing. AI makes the writing part dramatically more efficient.

The Grant Writing Workflow

Step 1: Find grants. Use Perplexity or Claude to research available funding:

Prompt: "What grants are available for [your description — e.g., emerging classical musicians, jazz composers, music educators, community arts organizations] in [your region/country] in 2026? For each, give me: the grant name, administering organization, award amount, eligibility requirements, typical deadline, and a direct link to the application page. Include both well-known national grants and smaller local/regional opportunities."

Tip: Verify every grant Claude mentions. Check that it still exists, is currently accepting applications, and that you meet the eligibility requirements.

Step 2: Analyze the application requirements.

Prompt: "I am applying for [grant name]. Here are the application instructions and evaluation criteria: [paste them]. Analyze these and tell me: (1) What are the reviewers really looking for? Read between the lines. (2) What are the most common mistakes applicants make? (3) What should I emphasize to be competitive? (4) Create an outline for my narrative statement that addresses every criterion."

Step 3: Draft your narrative.

Prompt: "Based on the outline we developed, draft my grant narrative. Here is the information about my project: [describe your project in detail — what it is, why it matters, who it serves, what makes it unique, your qualifications to do it]. Here is my budget: [describe]. Draft a compelling narrative that addresses every evaluation criterion. The tone

should be confident and clear — not academic, not sales-y. [Word limit] words."

Step 4: Review against the criteria.

Prompt: "Read my draft narrative and evaluate it against the grant's evaluation criteria. For each criterion, rate how well my narrative addresses it (strong/adequate/weak) and suggest specific improvements. Be harsh — I would rather fix weaknesses now than after submission."

Exercises

♦ **Exercise 1:** Research grants available to you right now. Use Claude and Perplexity to build a list of at least 10 opportunities. Create a spreadsheet with the grant name, deadline, award amount, and your eligibility status.

♦ **Exercise 2:** Choose one grant from your list and go through the full workflow: analyze the requirements, outline your response, draft the narrative, and review it against the criteria. Even if you do not submit, this exercise teaches you the grant-writing process.

Day 2: Financial Planning & Budgeting with AI

Financial literacy is not taught in most music programs, but it is essential for a sustainable career. AI can help you understand and manage your finances.

Setting Up Your Financial System

Prompt: "I am a freelance musician. Help me set up a simple financial tracking system. I need to track: income by source (teaching, performing, recording, commissions, grants), expenses by category (instrument maintenance, travel, marketing, professional development, supplies), tax obligations (quarterly estimated payments, deductions), and a simple budget. Recommend a free tool and walk me through setting it up step by step."

Tax Basics for Musicians

Prompt: "Explain the basics of taxes for a freelance musician in [your country/state]. Cover: what counts as taxable income, common deductions musicians can claim (instrument purchases, repairs, travel, home studio, music purchases, lessons, professional memberships), quarterly estimated tax payments, and what records I need to keep. I have never done freelance taxes before — explain everything from scratch."

Tip: AI can give you a good general understanding of taxes, but consult an accountant for your specific situation, especially as your income grows.

Exercises

♦ **Exercise:** Set up a basic financial tracking system using the AI's recommendations. Enter your income and expenses for the past month. Does the system work? What adjustments do you need?

Day 3: Contract Review & Negotiation Support

Build on what you learned in Week 2 about AI contract review. Today, go deeper:

Prompt: "I am negotiating a [type of contract — performance, recording, teaching, commission]. The offered rate is [amount]. Help me prepare for the negotiation: (1) What is the typical market rate for this type of work in [your region]? (2) What non-monetary benefits should I negotiate for? (3) Draft a professional counter-offer email that asks for [what you want] while maintaining a positive relationship. (4) What are common negotiation mistakes musicians make?"

Day 4: Networking & Relationship Management

Prompt: "I want to build stronger professional relationships in the music industry. I am currently connected with [describe your network — school colleagues, local musicians, a few industry contacts]. Help me create a networking strategy: (1) identify the types of people I should be connecting with for my career goals, (2) suggest specific outreach templates for cold introductions, warm follow-ups, and maintaining existing relationships, (3) recommend a simple system for tracking my professional relationships, and (4) give me a monthly networking 'to-do list' that takes no more than 2 hours per month."

Day 5: AI for Tour Planning & Logistics

Prompt: "I am planning a [solo tour / chamber music tour / series of performances] in [region] over [timeframe]. I have confirmed dates in [cities]. Help me with: (1) routing — the most efficient travel order, (2) a budget template covering travel, lodging, meals, and miscellaneous expenses, (3) a venue outreach email template for cities where I still need to book, (4) a packing checklist for a traveling musician, and (5) a promotional timeline — when to announce each date, when to do social media pushes."

Day 6: Staying Current — The AI Landscape Evolves

AI is changing faster than any technology in history. Tools you learn today may be obsolete in a year, and new tools will emerge that do things we cannot currently imagine. The most important skill is not mastering any single tool — it is developing the ability to evaluate and adopt new tools quickly.

Strategies for staying current:

- Follow AI news: Subscribe to newsletters like 'The Neuron,' 'Ben's Bites,' or 'AI for Creatives'
- Experiment monthly: Set aside one hour per month to try a new AI tool you have heard about
- Join communities: Reddit's r/artificial, music-specific AI communities, and your school's tech groups
- Attend workshops: Many schools and organizations now offer AI workshops for creative professionals
- Teach others: Teaching what you know is the best way to solidify your understanding

Day 7: Your AI-Powered Career Plan

Prompt: "I am a [your description] musician with [X] years until graduation (or at [career stage]). My goals for the next three years are: [list specific goals — concerts, recordings, degrees, positions, projects]. Based on everything I have learned about AI tools in this course, create a comprehensive plan for how AI should fit into my career over the next year: (1) daily AI habits I should maintain, (2) monthly AI tasks (like content creation, grant applications, financial review), (3) tools I should invest in (free vs. paid), (4) skills I should continue developing, and (5) specific AI-powered projects I should pursue to advance my goals."

Week 8: How the Web Works & Your Development Environment

In Week 6, you built a website and deployed it. That was a static site — files sitting on a server, the same for every visitor. This week, you go deeper. You are going to understand how the web actually works: how browsers talk to servers, how data flows, and how dynamic web applications are built. This conceptual foundation makes everything in Weeks 9 and 10 click into place.

Day 1: How the Internet Actually Works

The Big Picture

When you type a URL into your browser and press Enter, here is what happens, step by step:

Step 1: Your browser looks up the domain name. The URL 'yourname.com' is a human-readable address. Computers communicate using numerical IP addresses (like 172.67.182.31). Your browser asks a DNS (Domain Name System) server to translate the domain name into an IP address. DNS is like the internet's phone book.

Step 2: Your browser sends an HTTP request. HTTP (HyperText Transfer Protocol) is the language browsers and servers use to communicate. Your browser sends a request that says, essentially, 'Please send me the web page at this address.' This request travels across the internet to the server at that IP address.

Step 3: The server processes the request. If it is a static site (like the one you built in Week 6), the server simply finds the HTML file and sends it back. If it is a dynamic application (like a booking system or social media platform), the server runs code, possibly queries a database, and constructs a response on the fly.

Step 4: The server sends an HTTP response. This response contains the HTML, CSS, JavaScript, and any other files needed to display the page. It also includes a status code: 200 means 'success, here is your page,' 404 means 'that page does not exist,' 500 means 'something went wrong on the server.'

Step 5: Your browser renders the page. It reads the HTML (structure), applies the CSS (styling), and executes the JavaScript (behavior). The result is the visual web page you see.

This entire process — DNS lookup, HTTP request, server processing, response, rendering — happens in milliseconds. Every time you visit any website, this is what is happening.

Static vs. Dynamic

Static sites — The same content for every visitor. The server just sends pre-made files. Your musician website from Week 6 is static. Static sites are fast, simple, cheap to host, and perfect for portfolios, blogs, and informational sites.

Dynamic applications — Content changes based on who is visiting, what they do, or what data is in a database. When you log into Spotify, it shows your playlists — not everyone else's. That is dynamic. Dynamic apps need a backend: server-side code that processes requests and talks to a database.

Frontend vs. Backend

Frontend (client-side) — Everything that happens in the browser — the HTML, CSS, and JavaScript that the user sees and interacts with. Your Week 6 website was pure frontend.

Backend (server-side) — Code that runs on the server. It handles requests from the frontend, processes data, talks to databases, enforces security rules, and sends responses back. The user never sees backend code directly.

Full-stack — Frontend + backend together. A full-stack application has a user interface (frontend), server logic (backend), and usually a database for persistent storage.

APIs: How Frontend Talks to Backend

An API (Application Programming Interface) is a set of rules for how software components communicate. In web development, the frontend sends requests to the backend API, and the backend API sends back data — usually in a format called JSON.

Here is a real example. Imagine a practice log app. The frontend is the page you see — a form where you type what you practiced. When you click 'Save,' the frontend sends an HTTP request to the backend API: 'POST /api/practice-entries' with the data you typed. The backend receives this, saves it to a database, and sends back a response: 'Got it, saved successfully.' The frontend updates to show the new entry.

JSON (JavaScript Object Notation) looks like this:

```
{ "piece": "Bartok Sonata", "duration": 45, "date": "2026-04-15", "notes": "Worked on fugue rhythm" }
```

It is a simple, human-readable way to structure data. Every modern web application uses JSON to pass data between frontend and backend.

Exercises

♦ **Exercise 1:** Open your browser and visit your deployed website. Right-click anywhere and select 'Inspect' (or press F12). Click the 'Network' tab. Reload the page. You will see every HTTP request your browser made — each file it downloaded from the server. This

is the request/response cycle in action. Note how many files are loaded and what types they are.

♦ **Exercise 2:** Ask Claude at claude.ai: 'Explain the difference between a static website and a dynamic web application using a music analogy. Then give me five examples of each that a musician might encounter.' Write the best examples in your journal.

Day 2: The Terminal — Going Deeper

You used the terminal in Week 6 for basic navigation and Git commands. Today you go deeper — learning the commands you will need for application development.

Package Managers: npm

When you installed Node.js, you also got npm (Node Package Manager). npm lets you download and install packages — pre-written code that other developers have shared. Instead of writing everything from scratch, you use packages for common tasks: running a server, connecting to a database, handling security, and much more.

npm init -y — Creates a package.json file — the 'manifest' for your project. It lists your project's name, version, and all the packages it depends on. The '-y' flag accepts default settings.

npm install express — Downloads the 'express' package (a web server framework) and adds it to your project. It gets saved in a folder called 'node_modules' and listed in package.json.

npm install — Run this without a package name to install everything listed in an existing package.json. Useful when you download someone else's project — you run 'npm install' to get all the packages it needs.

node filename.js — Runs a JavaScript file. This is how you start your server: 'node server.js'.

The .gitignore File

Some files should not be pushed to GitHub. The node_modules folder (which can contain thousands of files from downloaded packages) should be excluded — anyone who clones your project can recreate it by running 'npm install.' Secret files like API keys should also be excluded.

Create a file called .gitignore in your project root. Tell Claude Code:

Prompt: "Create a .gitignore file for a Node.js project. Exclude node_modules, .env files, and any other common files that should not be in version control."

Running a Local Server

When you built your static site, you could open HTML files directly in your browser. For dynamic applications, you need a local server running on your computer. You will start a server with a command like 'node server.js,' and it will run at an address like 'http://localhost:3000.' 'localhost' means 'this computer,' and 3000 is the port number (think of it as a door the server listens on). You visit this address in your browser to see your app while developing.

Exercises

♦ **Exercise 1:** In your terminal, create a new folder called 'practice-app'. Navigate into it. Run 'npm init -y' to create a package.json. Open the package.json in VS Code and read it — Claude Code can explain each field if you ask.

♦ **Exercise 2:** In the same folder, run 'npm install express'. Notice the new node_modules folder (type 'ls' to see it). Open package.json again — 'express' now appears under 'dependencies.' This is how projects track what packages they use.

Day 3: Setting Up Your Development Environment

VS Code for Development

You installed VS Code in Week 6. Now set it up for application development. Open VS Code and install these extensions (click the four-squares icon in the left sidebar, search by name, and click Install):

- Live Server — Lets you preview web pages with auto-refresh when files change
- Prettier — Automatically formats your code to be clean and readable
- ESLint — Catches common JavaScript mistakes before they become bugs

To open your project in VS Code from the terminal, navigate to your project folder and type 'code .' (note the period — it means 'this folder'). VS Code opens with your project's file tree on the left.

The Development Workflow

Here is the workflow you will use for all application development going forward. Memorize this cycle:

- 1. Open your project folder in the terminal
- 2. Start Claude Code ('claude')
- 3. Describe what you want to build or change
- 4. Claude Code creates/edits files
- 5. Start your server ('node server.js') to see the result at localhost
- 6. Review in your browser — does it look and work correctly?

- 7. If not, tell Claude Code what to fix (you do not need to restart Claude Code)
- 8. When satisfied, save to Git: 'git add .', 'git commit -m "description"', 'git push'

Exercises

♦ **Exercise:** Set up VS Code with the three extensions. Open your practice-app folder in VS Code. Start Claude Code in that folder and ask it: 'Create a simple Express server in server.js that serves a web page saying Welcome to my Practice App. The page should have a clean design.' Start the server with 'node server.js' and visit <http://localhost:3000> in your browser.

Day 4: Claude Code — Your AI Development Partner (Deep Dive)

Working Effectively with Claude Code

You have been using Claude Code since Week 6. Now let us go deeper into the techniques that make it truly powerful as a development partner.

Technique 1: Describe the What, Not the How

You do not need to know programming to use Claude Code effectively. Describe what you want the application to do, not how to code it. Let Claude Code make the technical decisions.

Less effective: 'Create an Express route with a GET handler that queries SQLite...'

More effective: 'Add a page that shows all my practice sessions from the database, sorted by date with the most recent first.'

Technique 2: Iterate Visually

Look at the result in your browser after every change. Describe what you see that needs to change. 'The list is there but it looks ugly — add spacing between items, make the dates bold, and put each entry in a card with a subtle shadow.' You are the art director; Claude Code is the developer.

Technique 3: Ask Claude Code to Explain

After Claude Code builds something, ask: 'Explain what you just created. Walk me through the files and what each part does.' You do not need to understand every line, but knowing the general structure — which file does what, how data flows — helps you direct changes more effectively.

Technique 4: Use Claude Code for Debugging

When something goes wrong (and it will), do not panic. Tell Claude Code what happened: 'When I click the Save button, nothing happens. The page should add the entry to the list. Fix it.' Or: 'I see this error in my terminal: [paste the error message]. What is wrong and how do I fix it?' Claude Code can read error messages, diagnose problems, and fix them.

Exercises

♦ **Exercise:** Continue building the practice-app. Ask Claude Code to add: a form on the page where you can enter what piece you practiced, how long (in minutes), and any notes. When you submit the form, the entry should appear in a list below the form. For now, the entries do not need to be saved permanently (we will add a database tomorrow). Iterate on the design until it looks professional.

Day 5: Your First Web Page with Claude Code (Full Project)

Today you build a slightly more complex static project to solidify your understanding of HTML, CSS, and JavaScript before moving to full-stack applications next week.

Project: Interactive Concert Program

Ask Claude Code to build an interactive concert program page — something you could share as a link instead of (or alongside) a printed program.

Prompt: "Build an interactive concert program as a web page. The concert is [describe your concert — performer, date, venue, program].

The page should include:

- 1. An elegant header with the concert title, date, venue, and performer name*
- 2. Each piece on the program as a section with: title, composer, duration, and program notes (I will provide these)*
- 3. Clickable sections that expand to show the program notes and collapse to hide them*
- 4. A 'About the Performer' section at the bottom with my bio*
- 5. Clean, elegant styling that looks like a professional digital program*
- 6. Works perfectly on mobile since audience members will view it on their phones*

Use HTML, CSS, and JavaScript. Make the expand/collapse interactive."

Exercises

♦ **Exercise:** Build the interactive concert program. Add your real content. Deploy it to Vercel or Cloudflare Pages (you already know how from Week 6). Share the link with a friend and ask them to open it on their phone.

Day 6: Git & Version Control (Deep Dive)

Why Git Matters More Than You Think

You have been using Git to push code to GitHub. But Git is much more than a deployment tool — it is your safety net. Every time you commit, you save a snapshot of your entire project. If you make a change that breaks something, you can go back to any previous snapshot instantly. This is why professional developers commit frequently — it is like having unlimited undo.

The Git Workflow, Revisited

git status — Shows which files have changed since your last commit. Run this frequently — it tells you the current state of your project.

git add filename — Stages a specific file for the next commit. 'git add .' stages everything.

git commit -m 'message' — Saves a snapshot with a description. Write clear messages: 'Added contact form' is better than 'updates.'

git log — Shows the history of all commits. Each has a unique ID, a message, and a timestamp.

git diff — Shows exactly what changed in your files since the last commit — line by line.

git push — Uploads your commits to GitHub.

Commit Early, Commit Often

The golden rule: commit every time something works. Finished the navigation menu? Commit. Got the form working? Commit. Fixed a bug? Commit. Each commit is a safe point you can return to. If your next change breaks something, you are never more than one undo away from a working version.

Exercises

♦ **Exercise:** Go back to your musician website from Week 6. Run 'git log' to see all your commits. Run 'git diff' after making a small change to see what changed. Practice the full cycle: make a change, 'git status', 'git add .', 'git commit -m "message"', 'git push'. Do this five times with five different small changes to build muscle memory.

Day 7: Review & Confidence Building

This week you learned:

- How the internet works — DNS, HTTP, request/response, static vs. dynamic
- Frontend vs. backend vs. full-stack
- APIs and JSON — how the parts of a web application communicate
- npm — installing packages and managing dependencies
- The development workflow — Claude Code, local server, browser, iterate, commit
- Git at a deeper level — your safety net for all development work

♦ **Review Exercise:** In your journal, explain in your own words: (1) What happens when someone visits your website, (2) what the difference is between a static site and a dynamic application, (3) what an API does, and (4) why Git matters. If you can explain these four things clearly, you are ready for Week 9.

Week 9: Building Real Applications with Claude Code

This is the week where you go from building static pages to building dynamic applications — software that processes data, stores information, and does something useful. Claude Code does the heavy lifting of writing code. Your job is to architect: decide what the app should do, how it should look, and how the pieces fit together.

Day 1: How Web Applications Work

The Full Stack in Practice

A web application has three layers. Here is a musician-friendly way to think about them:

Frontend (the performance) — What the audience sees and hears. The web pages, buttons, forms, and visual design. Built with HTML, CSS, and JavaScript. This is the part that runs in the user's browser.

Backend (the backstage crew) — The logic that makes things happen behind the scenes. When a user submits a form, the backend receives the data, processes it (validates it, transforms it, decides what to do with it), and sends a response. Built with Node.js and a framework like Express.

Database (the filing cabinet) — Where data lives permanently. Practice logs, user accounts, event listings, contact form submissions — anything the app needs to remember between visits gets stored in a database.

How They Connect

Here is the full flow for a practice log application:

1. The user opens the app in their browser (frontend loads)
2. The user fills out a form: 'Practiced Bartók Sonata for 45 minutes'
3. The user clicks 'Save'
4. JavaScript on the frontend sends an HTTP POST request to the backend API
5. The backend receives the request, validates the data, and inserts it into the database
6. The backend sends a response: 'Saved successfully'
7. The frontend updates the page to show the new entry in the list
8. When the user visits the app tomorrow, the frontend asks the backend for all entries
9. The backend queries the database and sends back the list
10. The frontend displays all the entries — the data persisted between visits

REST APIs

Most web applications use REST (Representational State Transfer) — a convention for how the frontend and backend communicate. REST uses standard HTTP methods:

GET — Retrieve data. 'GET /api/practice-entries' → 'Send me all practice entries.'

POST — Create new data. 'POST /api/practice-entries' with data → 'Save this new entry.'

PUT — Update existing data. 'PUT /api/practice-entries/42' → 'Update entry number 42 with this new data.'

DELETE — Remove data. 'DELETE /api/practice-entries/42' → 'Delete entry number 42.'

These four operations — Create, Read, Update, Delete — are called CRUD, and they cover almost everything a web application does with data.

Exercises

♦ **Exercise:** Ask Claude at claude.ai: 'I want to build a Gig Tracker app where I can add upcoming gigs with venue, date, pay, and notes. Walk me through the complete architecture: what is the frontend, what is the backend, what is the database, and what API endpoints do I need? Use the GET/POST/PUT/DELETE convention and explain each one.' Save this as your reference for the rest of the week.

Day 2: Your First Backend — Node.js & Express

What Express Is

Express is a framework for Node.js that makes it simple to create a web server — the backend that receives requests and sends responses. It is the most popular Node.js framework and is used by companies of all sizes. Claude Code knows Express inside and out.

Building Your Backend

Create a new folder for your project (or use the practice-app from last week). Start Claude Code and tell it:

Prompt: "Create a backend for a Practice Log application using Node.js and Express. The server should:

- 1. Serve a frontend page at the root URL (/)*
- 2. Have an API endpoint: GET /api/entries — returns all practice entries as JSON*
- 3. Have an API endpoint: POST /api/entries — accepts a new practice entry (piece, duration in minutes, notes, date) and saves it*

4. For now, store entries in memory (a JavaScript array) — we will add a real database tomorrow
5. The frontend should have a form to add entries and a list showing all existing entries
6. Use 'fetch' on the frontend to communicate with the API
7. Make it look clean and professional

Set up the project with npm init and install Express."

Claude Code will create a server.js file (the backend), an HTML file (the frontend), and a package.json. When it is done, start the server:

```
node server.js
```

Visit <http://localhost:3000> in your browser. You should see your practice log app. Try adding entries using the form. They will appear in the list. Reload the page — the entries are still there (as long as the server is running). Stop the server with Ctrl+C and restart it — the entries are gone, because they were only stored in memory. Tomorrow, you will fix this with a database.

Understanding What Claude Code Built

Ask Claude Code: 'Explain the server.js file. Walk me through what each part does — the imports, the routes, how the frontend communicates with the backend.'

You do not need to memorize the code. But understanding the structure — this part handles GET requests, this part handles POST requests, this is where data gets processed — makes you a better director of the AI development process.

Exercises

♦ **Exercise 1:** Build the practice log app as described above. Add at least five practice entries through the form. Verify they appear in the list.

♦ **Exercise 2:** Ask Claude Code to add a feature: 'Add a DELETE button next to each practice entry that removes it from the list and the server.' Test it. Then ask Claude Code to explain how the delete functionality works end-to-end.

Day 3: Databases — Storing Real Data

Why You Need a Database

Yesterday your practice entries disappeared when you restarted the server. That is because they were stored in the server's memory, which is wiped clean every time the server stops. A database solves this — it stores data on disk, permanently, so it survives restarts, crashes, and even moving to a different server.

SQLite: The Perfect Starter Database

SQLite is a database that stores everything in a single file on your computer. It is the most widely deployed database in the world (it is in your phone, your browser, and many of the apps you use daily). It requires no setup, no server, no configuration — just install a package and start using it. This makes it ideal for learning and for small to medium applications.

SQL: The Language of Databases

SQL (Structured Query Language) is how you talk to relational databases. It reads almost like English. Here are the four essential commands:

```
CREATE TABLE entries (id INTEGER PRIMARY KEY, piece TEXT, duration
INTEGER, notes TEXT, date TEXT);
```

- Creates a table (like a spreadsheet) with columns for each piece of data.

```
INSERT INTO entries (piece, duration, notes, date) VALUES ('Bartok Sonata', 45,
'Worked on fugue', '2026-04-15');
```

- Adds a new row of data.

```
SELECT * FROM entries ORDER BY date DESC;
```

- Retrieves all rows, sorted by date (newest first). The * means 'all columns.'

```
DELETE FROM entries WHERE id = 3;
```

- Removes the entry with id 3.

Claude Code handles all of this for you, but knowing what SQL looks like helps you understand what your application is doing and gives you the vocabulary to ask for what you want.

Adding a Database to Your App

Start Claude Code in your practice-app folder and tell it:

Prompt: "Add a SQLite database to the practice log app. Use the 'better-sqlite3' npm package. Replace the in-memory array with a real database that:

- 1. Creates a 'entries' table if it does not exist (piece TEXT, duration INTEGER, notes TEXT, date TEXT)*
- 2. Saves new entries to the database when POST /api/entries is called*
- 3. Reads all entries from the database when GET /api/entries is called*
- 4. Deletes entries from the database when DELETE /api/entries/:id is called*

*5. The database file should be called 'practice.db'
Keep everything else the same — the frontend should work exactly as before."*

After Claude Code makes the changes, restart your server. Add some entries. Stop the server (Ctrl+C) and start it again. Your entries are still there. That is the power of a database — your data persists.

Exercises

♦ **Exercise 1:** Add the database as described above. Verify data persists after restarting the server. Add ten entries, restart, confirm all ten are still there.

♦ **Exercise 2:** Ask Claude Code: 'Show me the contents of the database directly. What is in the entries table right now?' Claude Code can query the database and show you the raw data. This helps you understand what is actually being stored.

Day 4: Connecting Frontend to Backend

The fetch API

The 'fetch' function is how JavaScript in the browser (frontend) communicates with your server (backend). When your practice log app sends a new entry to the server, it uses fetch. When it loads existing entries, it uses fetch. Understanding fetch — even at a high level — helps you direct Claude Code more effectively.

Here is what a fetch call looks like in plain English: 'Send a POST request to /api/entries with this JSON data, and when you get a response, update the page.' You do not need to write this code yourself — Claude Code writes it — but knowing the pattern helps you ask for what you want.

Building a Better Frontend

Your practice log app works, but the frontend is probably basic. Today, make it great. Tell Claude Code:

Prompt: "Redesign the practice log app frontend. Keep all the same functionality but make it look professional and polished:

- 1. Add a header with the app name and a clean logo/icon*
- 2. Style the form with proper input fields, labels, and a styled submit button*
- 3. Display entries as cards with the piece name as a heading, duration and date as subtext, and notes in a lighter font*
- 4. Add the total practice time displayed prominently (sum of all entry durations)*
- 5. Add filtering — buttons to show 'Today,' 'This Week,' and 'All Time'*

6. *Make it fully responsive for mobile*
7. *Use a color scheme of [your preferred colors]"*

Exercises

♦ **Exercise:** Redesign the frontend as described above. Iterate with Claude Code until it looks like a professional app you would actually want to use. Show it to a friend and ask for feedback. Fix any issues.

Day 5: API Keys, Environment Variables & Security

What API Keys Are

An API key is a password that identifies your application to an external service. When your app uses the Claude API, Spotify API, Google Maps, or any external service, you need an API key. The service uses this key to know who is making the request and to track your usage.

API keys are secrets. If someone gets your API key, they can make requests pretending to be you — potentially running up charges on paid services. Protecting API keys is one of the most important security practices in development.

Environment Variables

The golden rule: never put API keys, passwords, or secrets directly in your code. If your code is on GitHub (which it needs to be for deployment), anyone can see it. Instead, store secrets in environment variables — values that live outside your code, in the server's configuration.

In development (on your computer), you use a `.env` file. This is a simple text file that lists your secrets:

```
ANTHROPIC_API_KEY=sk-ant-abc123...  
DATABASE_URL=postgresql://user:pass@host/db  
MY_SECRET_VALUE=something-private
```

Your `.gitignore` file (which you created earlier) must include `.env` so this file never gets pushed to GitHub. Claude Code knows this and will set it up correctly if you ask.

In production (on Vercel, Railway, or any deployment platform), you set environment variables in the platform's dashboard. We will cover this in Week 10.

Using the dotenv Package

The 'dotenv' package loads your `.env` file into your application. Ask Claude Code:

Prompt: "Add dotenv to the project so I can use environment variables. Create a .env file with a placeholder API key. Make sure .env is in the .gitignore. Show me how the server reads the variable."

Security Basics

Here are the security principles every developer — even an AI-assisted one — should know:

Never commit secrets — API keys, passwords, database credentials must never appear in code that gets pushed to GitHub. Always use environment variables and .env files.

Validate user input — Never trust data from users. If your form accepts a 'duration' field, make sure it is actually a number before saving it. Claude Code handles this if you tell it to.

Use HTTPS — All communication between browsers and your server should be encrypted. Vercel and Cloudflare provide HTTPS automatically — one of the reasons we use them.

Keep dependencies updated — The packages you install (Express, better-sqlite3, etc.) occasionally have security patches. Run 'npm audit' periodically to check for known vulnerabilities.

Principle of least privilege — Only give your application the access it actually needs. If your app only reads data from an API, use a read-only API key if available.

Exercises

♦ **Exercise 1:** Add dotenv to your practice log app. Create a .env file with a variable called APP_NAME set to 'My Practice Log.' Have the server read this variable and display it as the app title. Verify that .env is in your .gitignore.

♦ **Exercise 2:** Run 'git status' and confirm that .env does NOT appear in the list of tracked files. If it does, you have a security problem — ask Claude Code to fix your .gitignore and remove .env from Git tracking.

♦ **Exercise 3:** Ask Claude Code: 'Review my application for security issues. Check that no secrets are in the code, user input is validated, and the .gitignore is correct.' Let it do a security audit.

Day 6: Building a Complete Application

Capstone Mini-Project: Setlist Manager

Today you build a more ambitious application to solidify everything you have learned. The Setlist Manager lets you create setlists for your performances — add pieces with metadata, reorder them, and see the total duration.

Prompt: "Build a Setlist Manager application with these features:

BACKEND (Node.js + Express + SQLite):

- Database tables: 'setlists' (id, name, date, venue) and 'pieces' (id, setlist_id, title, composer, duration_minutes, key, position, notes)*
- Full CRUD API for setlists and pieces*
- An endpoint to reorder pieces within a setlist*
- Total duration calculation*

FRONTEND:

- Home page showing all setlists as cards*
- Click a setlist to see its pieces in order*
- Add/edit/delete pieces with a form*
- Drag-and-drop or up/down buttons to reorder pieces*
- Total duration displayed prominently*
- Clean, professional design*
- Responsive for mobile*

Use dotenv for any configuration. Include a .gitignore."

This is a real, useful application. You could genuinely use this for your performances. Iterate with Claude Code until it works well and looks professional.

Exercises

♦ **Exercise:** Build the Setlist Manager. Create at least two setlists with real pieces from your repertoire. Test all features: adding, editing, deleting, reordering. Push to GitHub.

Day 7: Testing, Debugging & Code Review

When Things Go Wrong

Bugs are a normal part of development — something works in one browser but not another, data saves incorrectly, a button does nothing when clicked. The skill is not avoiding bugs (impossible) but diagnosing and fixing them quickly.

Reading Error Messages

When something goes wrong, your terminal or browser console will usually show an error message. These messages are intimidating at first but contain valuable information. The most important parts are:

- The error type (e.g., 'TypeError', 'ReferenceError', 'SyntaxError')
- The message (e.g., 'Cannot read property x of undefined')
- The file and line number where the error occurred

Copy the entire error message and paste it to Claude Code: 'I am getting this error: [paste error]. What is wrong and how do I fix it?' Claude Code can almost always diagnose and fix the issue immediately.

Browser Developer Tools

Right-click anywhere on your page and select 'Inspect' (or press F12). The Console tab shows JavaScript errors from the frontend. The Network tab shows HTTP requests and responses between your frontend and backend. These are your windows into what your application is actually doing.

Asking Claude Code for a Code Review

Prompt: "Review the entire codebase for this application. Look for: bugs, security issues, performance problems, and code quality issues. List everything you find and fix the important ones."

Exercises

♦ **Exercise:** Ask Claude Code to review your Setlist Manager code. Read its findings. Let it fix any real issues. Then test the application thoroughly — try every feature, try unexpected inputs, try it on mobile.

Week 10: Deployment & Going Live

You have built a real application with a frontend, backend, and database. Now you make it accessible to the world. This week covers everything needed to take an application from your computer to a live server.

Day 1: Understanding Servers & Hosting

What a Server Actually Is

A server is a computer that is always on and always connected to the internet, waiting for requests. When you ran 'node server.js' on your laptop, your laptop was the server — but only you could access it (via localhost). To make your app available to the world, you need it running on a server that anyone can reach.

You do not buy or manage a physical server. Modern hosting platforms give you a virtual server in the cloud. You upload your code, they run it, and they handle the hardware, networking, and uptime.

Types of Hosting

Static hosting (Vercel, Cloudflare Pages, Netlify, GitHub Pages) — For static sites — HTML, CSS, and JavaScript files served as-is. This is what you used in Week 6 for your musician website. Free, fast, and simple. But it cannot run a backend or database.

Application hosting (Railway, Render, Fly.io) — For full-stack applications — your Node.js/Express backend, database, and frontend all running together. These platforms run your server code 24/7. Most have free or low-cost starter tiers.

Serverless functions (Vercel Functions, Cloudflare Workers) — A middle ground. Your backend code runs on-demand (when someone makes a request) rather than 24/7. Cheaper for low-traffic apps. More complex to set up but very scalable.

Platform Comparison for Musicians

Railway (railway.app) — The simplest way to deploy a full-stack app. Connects to GitHub, auto-detects Node.js, deploys in minutes. Includes built-in PostgreSQL databases. Free tier with \$5 of monthly usage. Excellent for learning and small projects.

Render (render.com) — Similar to Railway. Free tier for web services (with a spin-down after inactivity — your app may take 30 seconds to 'wake up' on the first visit). Free PostgreSQL databases for 90 days. Very beginner-friendly.

Fly.io — More powerful but slightly more complex. Deploys your app in data centers around the world for fast access globally. Free tier available. Better for apps you want to scale.

Exercises

♦ **Exercise:** Create free accounts on Railway (railway.app) and Render (render.com). Explore their dashboards. Read their 'Getting Started' guides. You will use one of these tomorrow.

Day 2: Deploying a Static Site (Review)

You already deployed a static site in Week 6. Today, quickly review the process and ensure your musician website is in good shape on Vercel or Cloudflare Pages. Make any updates you have been wanting to make, push them to GitHub, and verify the live site updates automatically.

If you have not already, use this day to connect a custom domain to your musician website. The process was covered in Week 6, Day 6.

Exercises

♦ **Exercise:** Update your musician website with any improvements. Push to GitHub. Verify the live site updates. If you have a custom domain, test it. If not, consider buying one today.

Day 3: Deploying a Full-Stack Application

Deploying to Railway (Step by Step)

Let us deploy your Setlist Manager (or Practice Log) application to Railway. This is a full-stack app with a backend and database, so it requires an application hosting platform rather than static hosting.

Step 1: Make sure your app is pushed to GitHub. Your latest code should be in your GitHub repository. Run 'git status' to check for uncommitted changes, commit them, and push.

Step 2: Prepare your app for deployment. Ask Claude Code:

Prompt: "Prepare this application for deployment on Railway. Make sure:

- 1. The server reads the PORT from an environment variable (Railway assigns its own port)*
- 2. The start command is defined in package.json (under 'scripts': {'start': 'node server.js'})*
- 3. A .gitignore is in place excluding node_modules and .env*
- 4. All dependencies are listed in package.json"*

Step 3: Go to railway.app and sign in. Click 'New Project' → 'Deploy from GitHub Repo.' Select your application's repository.

Step 4: Railway auto-detects that this is a Node.js application. It will run 'npm install' to install your dependencies and then 'npm start' to run your server.

Step 5: Wait 2-3 minutes for the build. Railway shows you real-time logs — you can watch your server starting up, just like you see in your local terminal.

Step 6: Click 'Settings' → 'Generate Domain' to get a public URL. Your app is now live.

Deploying to Render (Alternative)

If you prefer Render:

1. Go to render.com and sign in.
2. Click 'New' → 'Web Service.'
3. Connect your GitHub account and select your repository.
4. Set the build command to 'npm install' and the start command to 'node server.js.'
5. Choose the free tier.
6. Click 'Create Web Service.' Your app deploys in 3-5 minutes.
7. Render gives you a URL like 'your-app.onrender.com.'

Exercises

♦ **Exercise:** Deploy your full-stack application to Railway or Render. Visit the live URL in your browser. Test all features — adding data, viewing data, deleting data. It should work exactly like it did on localhost. If something breaks, check the deployment logs and ask Claude Code for help.

Day 4: Databases in Production

SQLite vs. PostgreSQL in Production

Your local app uses SQLite, which stores data in a single file. This works well for development and for small applications, but has a limitation in production: most hosting platforms use ephemeral file systems, meaning files can be reset when the app redeploys. Your database file could disappear.

The solution: use a hosted database service. The most common is PostgreSQL, a powerful open-source database used by companies from startups to enterprises. Both Railway and Render offer PostgreSQL databases on their platforms.

Adding a PostgreSQL Database on Railway

Step 1: In your Railway project, click 'New' → 'Database' → 'PostgreSQL.'

Step 2: Railway creates a database and automatically adds a DATABASE_URL environment variable to your web service.

Step 3: Ask Claude Code to update your app:

Prompt: "Migrate this application from SQLite to PostgreSQL. Use the 'pg' npm package. Read the database connection string from the DATABASE_URL environment variable. Keep the same table structure and API behavior. Create the tables automatically if they do not exist. Keep SQLite as a fallback for local development (if DATABASE_URL is not set, use SQLite; if it is set, use PostgreSQL)."

Step 4: Push the updated code to GitHub. Railway automatically redeploys with the new database connection.

Environment Variables in Production

On your computer, secrets live in your .env file. In production, you set them in your hosting platform's dashboard. In Railway: click your service → 'Variables' tab → add key-value pairs. In Render: click your service → 'Environment' tab.

Your DATABASE_URL is already set automatically by Railway when you add a PostgreSQL database. For other secrets (like API keys for external services), you add them manually in the same Variables/Environment section.

Exercises

♦ **Exercise:** If you are on Railway, add a PostgreSQL database and migrate your app. If you are on Render (which also offers free PostgreSQL), do the same. Test that data persists after redeployment — add entries, push a small code change, wait for redeployment, verify your entries are still there.

Day 5: Custom Domains, HTTPS & DNS

How Custom Domains Work for Applications

The process for connecting a custom domain to your full-stack app is essentially the same as for your static site (Week 6). In Railway or Render, go to your service settings, find the 'Custom Domain' section, enter your domain, and update your DNS settings at your domain registrar.

HTTPS: Why It Matters

HTTPS (the padlock icon in your browser's address bar) means the connection between the user's browser and your server is encrypted. No one can intercept the data in transit. This is essential if your app handles any personal data (contact forms, login systems,

payment info). Vercel, Cloudflare, Railway, and Render all provide HTTPS automatically for custom domains — you do not need to set it up yourself.

Exercises

♦ **Exercise:** If you have a custom domain and want to point it to your application (rather than your static site), follow the platform's instructions. If not, your `.railway.app` or `.onrender.com` URL works perfectly. The important thing is that HTTPS is working — check for the padlock icon.

Day 6: Monitoring, Logs & Maintenance

Monitoring: Is Your App Running?

Once your app is live, you need to know if it goes down. Both Railway and Render have dashboards that show your app's status. You can also set up simple monitoring:

- Railway: The dashboard shows real-time status, CPU and memory usage
- Render: The dashboard shows deploy status and health checks
- Free monitoring services like UptimeRobot (uptimerobot.com) will ping your app every 5 minutes and email you if it goes down

Reading Logs

Logs are the messages your server prints while running. When something goes wrong, logs are where you find out why. Both Railway and Render show logs in real-time in their dashboards. When a user reports a problem, check the logs first.

Updating Your Live App

Updating your deployed app is the same process every time:

1. Make changes locally using Claude Code
2. Test locally (`node server.js`, check in browser)
3. `git add .`, `git commit -m 'description'`, `git push`
4. Your hosting platform automatically redeploys
5. Check the live site to verify the changes

If a deploy breaks something, you can roll back. Both Railway and Render keep your previous deployments and let you instantly revert to the last working version.

Exercises

♦ **Exercise 1:** Set up a free UptimeRobot account and add monitoring for your deployed application. It will email you if your app goes down.

♦ **Exercise 2:** Practice the update cycle: make a small visual change using Claude Code, test locally, push to GitHub, and verify the live site updates. Do this three times to build confidence in the deployment workflow.

Day 7: Your Complete Development Workflow

You now have a complete, professional development workflow:

- Design and plan with Claude at claude.ai (big-picture decisions, architecture)
- Build and code with Claude Code in the terminal (implementation)
- Test locally at localhost in your browser (verification)
- Save progress with Git (safety net, history)
- Push to GitHub (code storage, deployment trigger)
- Deploy automatically to Railway/Render (your app goes live)
- Monitor with logs and UptimeRobot (peace of mind)

This is the same workflow used by professional developers at companies of all sizes. The difference is that your AI partner handles the code while you handle the vision, design, and decision-making. That is a powerful combination.

Week 10 Reflection

Three weeks ago, you had never opened a terminal. Now you have built and deployed a full-stack web application with a backend, database, and professional hosting. You understand how the internet works, how data flows between frontend and backend, how to protect API keys, and how to keep a live application running. These are skills that most musicians — most people — do not have. They will serve you for the rest of your career.

Week 11: Capstone Projects

This final week brings everything together into a single project that demonstrates your AI fluency and produces something genuinely valuable for your career.

Day 1: Choosing Your Capstone

Choose one of the following capstone projects, or propose your own:

Capstone A: Complete Concert Production Package

Using AI tools, create everything needed for a professional concert: program notes for every piece, a concert poster, social media promotion (10 posts), a press release, spoken introductions, a concert recording plan, and a post-concert email to attendees. Every element should be polished and ready to use.

Capstone B: Complete Digital Presence Overhaul

Build or completely redesign your professional website, create a full EPK, develop a three-month content strategy with sample content for each month, design a brand identity guide, set up an email newsletter with the first three issues drafted, and create a social media calendar for 30 days of content.

Capstone C: AI-Powered Teaching Suite

Create a comprehensive teaching resource using AI: a full semester curriculum for a course you could teach, five complete lesson plans with materials, a student assessment rubric, a set of custom exercises and etudes, practice guides for three different levels, and a student-facing website or resource page.

Capstone D: Grant Application Portfolio

Research and prepare applications for three different grants. For each: analyze the requirements, write the narrative statement, develop the budget and budget justification, draft all supplementary materials, and create a submission checklist. The goal is three submission-ready applications.

Capstone E: Music Career Launch Kit

For a student about to graduate: Create a comprehensive career launch package including three versions of your bio, an artist statement, a complete EPK, a website, a financial plan for your first year, five tailored cover letters for different opportunity types (orchestra audition, teaching position, freelance pitch, residency application, festival submission), and a twelve-month career plan.

Capstone F: Custom Web Application

Build and deploy a full-stack web application that solves a real problem in your musical life. Ideas: a student studio management app (scheduling, assignments, progress tracking), a repertoire database with performance history and notes, a collaborative setlist planner for your ensemble, or an AI-powered practice coach that analyzes your log entries and suggests focus areas. Use Claude Code for development, deploy on Railway or Render with a PostgreSQL database and a custom domain.

Days 2-6: Planning, Building & Polishing

Day 2: Plan your capstone in detail. Use Claude to help create a project plan with daily milestones.

Days 3-5: Build. Dedicate each session to completing specific milestones. Use every AI skill you have learned.

Day 6: Polish. Review every element for quality, consistency, and professionalism. Ask Claude to review your work critically.

Day 7: Reflection & What Comes Next

You have completed an eleven-week course in AI literacy, creation, and application development. Take today to reflect on how far you have come.

Reflection Prompts

- Look at your Day 1 journal entry. How has your understanding of AI changed?
- What was the most useful tool or technique you learned? Why?
- What surprised you most about what AI can (or cannot) do?
- How has AI changed your daily routine as a musician?
- What ethical principles guide your use of AI?
- What do you want to learn next?

What Comes Next

Your AI education does not end here. It evolves with the technology. Here is how to keep growing:

- Continue using AI daily — fluency comes from consistent practice, just like music
- Explore additional courses on the Anthropic Academy (anthropic.skilljar.com) — new courses are added regularly
- Teach someone else what you learned — there is no better way to deepen your understanding
- Stay current with new tools and developments — set a monthly 'AI exploration' date
- Build more complex projects — each project teaches you something new

- Advocate for AI literacy in your music school — many of your colleagues need this knowledge

Part III — Appendices

Appendix A — The Complete AI Prompt Library for Musicians

This library contains ready-to-use prompts organized by category. Copy them, customize the bracketed sections, and use them as starting points. Each prompt has been tested and refined for quality output.

Performance & Preparation

Practice Plan Generator

Prompt: "Create a detailed [timeframe] practice plan for [piece] by [composer]. I have [hours] per day available. My current challenges are [specific challenges]. Include daily goals, weekly milestones, technique-specific exercises, and a strategy for performance readiness. Format as a table with columns for Day, Time Allocation, Focus Area, Specific Tasks, and Success Criteria."

Performance Anxiety Coach

Prompt: "I have a performance in [timeframe] and I am experiencing anxiety about [specific concerns]. Give me: (1) a daily mental preparation routine for the week before the performance, (2) a pre-performance ritual for the day of, (3) specific cognitive techniques for managing anxiety during performance, (4) a post-performance recovery routine. Base your advice on sports psychology and performance science, not generic reassurance."

Interpretation Guide

Prompt: "Help me develop an interpretation of [piece] by [composer]. I want to make thoughtful, informed choices about tempo, dynamics, phrasing, and character for each section. For each major section of the piece, suggest: the general character, tempo considerations, dynamic plan, key expressive moments, and how this section relates to the whole. Ground your suggestions in the score's markings, the historical context, and the performance tradition."

Audition Preparation

Prompt: "I am preparing for [audition type] in [timeframe]. The required repertoire is [list excerpts/pieces]. Create a comprehensive preparation plan that covers: technical preparation for each excerpt, musical interpretation notes, a mock audition schedule (when and how to simulate audition conditions), mental preparation strategies, and a checklist for audition day."

Sight-Reading Improvement

Prompt: "Design a 30-day sight-reading improvement program for [instrument] at an [intermediate/advanced] level. Each day should include: what to sight-read (suggest specific types of music or resources), how long, what to focus on, and a self-evaluation method. Include progressive difficulty and weekly checkpoints."

Writing & Communication

Program Notes

Prompt: "Write [length]-word program notes for [piece] by [composer] for an audience of [describe audience]. Include historical context, structural overview, and listening guidance. The tone should be [warm/academic/conversational/enthusiastic] — imagine you are [a friend explaining over coffee / a professor giving a pre-concert talk / a music critic writing for a magazine]. Avoid jargon unless you define it."

Grant Narrative

Prompt: "I am applying for [grant name] for my project: [describe project]. The evaluation criteria are [list criteria]. Write a [word count]-word narrative that addresses every criterion. Emphasize: [key strengths]. The tone should be confident, specific, and compelling — show do not tell. Include concrete details, not vague promises."

Professional Bio

Prompt: "Write a [long/medium/short] professional musician bio for [name]. Key information: [instrument, education, highlights, current activities, something distinctive]. Third person. Avoid: 'passionate,' 'dedicated,' 'captivating.' Start with something more interesting than my name. Tone: [confident but not arrogant / warm and approachable / serious and distinguished]."

Cold Outreach Email

Prompt: "Write an email from me ([your description]) to [recipient and their role] about [purpose]. Be concise (under [word count] words), professional but personable, and end with a clear, specific call to action. Do not be obsequious or use filler. Include a subject line that would get opened."

Thank You / Follow-Up

Prompt: "Write a thank-you email to [person] after [event/meeting]. Mention: [specific positive detail from the interaction]. Express interest in [future possibility] without being pushy. Keep it under [word count] words. Warm, professional tone."

Social Media Post

Prompt: "Write a [platform] post about [topic]. Audience: [describe]. Goal: [inform/promote/engage/inspire]. Include a hook in the first line, keep it under [character/word limit], and suggest 5-10 relevant hashtags. Tone: [describe your brand voice]."

Newsletter

Prompt: "Write an email newsletter to my subscriber list. Theme: [topic]. Include: a personal greeting, one main story or update, a call to action, and a sign-off. Length: [word count]. Tone: like a letter from a friend who also happens to be a musician, not a corporate email."

Research & Learning

Concept Explainer

Prompt: "Explain [music concept] as if tutoring me one-on-one. Start simple, then add complexity. Use examples from standard repertoire for [instrument]. Define technical terms in context. After explaining, give me 3 practice problems and check my work."

Score Analysis

Prompt: "Analyze [piece], movement [X], measures [Y-Z]. For each measure: identify the harmony, explain its function, describe the voice leading, and note anything unusual. Then: identify the three most interesting harmonic moments and explain why they are effective."

Composer Research

Prompt: "Give me a comprehensive overview of [composer]'s life and work. Include: biographical context (era, influences, personal circumstances), musical style (what makes their music distinctive), major works (with dates and significance), influence on later music, and recommended recordings for a newcomer to their work."

Recording Comparison

Prompt: "I am studying [piece]. Compare these recordings: [list 2-4 performers/recordings]. For each: describe their interpretive approach, notable tempo and dynamic choices, strengths, and what I can learn from their performance. Recommend which to use as a primary reference and why."

Teaching

Lesson Plan

Prompt: "Create a [duration]-minute lesson plan for a [age/level] [instrument] student working on [repertoire] who is struggling with [challenge]. Include: warm-up (with specific exercises), technique work (targeting the challenge), repertoire practice (specific passages and strategies), and a creative/fun activity. Time each section."

Curriculum Design

Prompt: "Design a [duration] curriculum for [course description]. Students are [level/background]. We meet [frequency]. Create a week-by-week syllabus with: topics, learning objectives, activities, assignments, and assessment criteria."

Exercise Generator

Prompt: "Create [number] progressive exercises for [technique] on [instrument], beginner to [target level]. For each: notation or description, purpose, common mistakes to watch for, and mastery criteria."

Student Evaluation

Prompt: "Write a semester evaluation for [student name], [level] [instrument] student. Repertoire covered: [list]. Strengths: [list]. Areas for improvement: [list]. Goals for next semester: [list]. Tone: encouraging, specific, honest. [Word count] words."

Business & Career

Invoice Generator

Prompt: "Create a professional invoice template for my freelance music work. Include fields for: my business name and contact info, client name and contact, date of service, description of service, rate, total amount, payment terms, and payment methods accepted. Format it cleanly."

Contract Analysis

Prompt: "I received a contract for [type of engagement]. [Paste contract]. Summarize key terms in plain English, flag concerning clauses, identify missing protections, and suggest questions to ask before signing."

Rate Calculator

Prompt: "Help me determine fair rates for my freelance music services. I offer: [list services — teaching, performing, recording, arranging, etc.]. I am based in [location] with [X years] experience. Research and suggest: hourly/per-service rates, how to account for preparation time and travel, when to negotiate higher, and how to communicate rates professionally."

Appendix B — AI Tool Directory & Comparison Charts

Text AI

Tool	URL	Best For	Pricing
Claude (Anthropic)	claude.ai	Long-form writing, analysis, creative work, file analysis	Free: generous limits Pro: \$20/mo
ChatGPT (OpenAI)	chatgpt.com	General purpose, image generation, web browsing, voice	Free: limited Plus: \$20/mo
Gemini (Google)	gemini.google.com	Google integration, research, multimodal (video/audio)	Free: generous Advanced: \$20/mo
Perplexity	perplexity.ai	Research with cited sources, fact-checking	Free: generous Pro: \$20/mo

Audio & Music AI

Tool	URL	Best For	Pricing
Moises.ai	moises.ai	Stem separation, chord detection	Free: 3-5 tracks/mo Premium: \$4/mo
LALAL.AI	lalal.ai	Stem separation (alternative to Moises)	Free: limited Pro: from \$15
Suno	suno.com	AI music generation from text	Free: limited generations Pro: \$10/mo
Udio	udio.com	AI music generation from text	Free: limited Standard: \$10/mo

LANDR	landr.com	AI mastering, distribution	Free: preview only Plans from \$6/mo
ElevenLabs	elevenlabs.io	Voice synthesis, voice cloning	Free: limited characters From \$5/mo
Descript	descript.com	Audio/video editing via text	Free: limited Pro: \$24/mo
BandLab	bandlab.com	Free music creation, AI mastering	Free

Image & Video AI

Tool	URL	Best For	Pricing
ChatGPT + DALL-E	chatgpt.com	Image generation integrated with text AI	Included with ChatGPT plans
Midjourney	midjourney.com	High-quality artistic image generation	From \$10/mo
Canva AI	canva.com	Image generation + graphic design	Free tier generous Pro: \$13/mo
Adobe Firefly	firefly.adobe.com	Commercially safe image generation	Free: limited Creative Cloud
Ideogram	ideogram.ai	Image generation with legible text	Free: limited Plans from \$7/mo
Runway	runwayml.com	AI video generation	Free: limited From \$12/mo
Pika	pika.art	AI video generation	Free: limited From \$8/mo

Productivity & Business AI

Tool	URL	Best For	Pricing
Notion	notion.so	Notes, databases, project	Free: personal

		management, AI writing	Plus: \$10/mo
Airtable	airtable.com	Spreadsheet-database for tracking	Free: limited Plus: \$20/mo
Zapier	zapier.com	Automation between apps	Free: 5 automations From \$20/mo
Gamma	gamma.app	AI presentations and documents	Free: limited Plus: \$8/mo
Calendly / Cal.com	cal.com	Scheduling automation	Free tiers available
Mailchimp	mailchimp.com	Email newsletters	Free: up to 500 subscribers

Development & Deployment

Tool	URL	Best For	Pricing
Claude Code	claude.ai/code	AI development partner in the terminal	Check current pricing at claude.ai
VS Code	code.visualstudio.com	Free code editor	Free
Node.js	nodejs.org	JavaScript runtime for backend development	Free
Git / GitHub	github.com	Version control and code hosting	Free
Vercel	vercel.com	Static site and serverless deployment	Free tier generous
Cloudflare Pages	pages.cloudflare.com	Static site deployment with CDN	Free tier generous
Railway	railway.app	Full-stack app and database hosting	Free: \$5/mo credit
Render	render.com	Full-stack app and	Free tier available

		database hosting	
--	--	------------------	--

Appendix C — Troubleshooting & FAQ

Claude gives me generic responses

You are not providing enough context. Go back to Day 3's lesson on the five elements of prompting. Add Role, Context, and Constraints to your prompt. The more specific you are, the more specific the response.

Claude says it cannot do something I think it should be able to do

Try rephrasing your request. Some wordings trigger safety filters that do not apply to your actual use case. Also, try breaking the task into smaller steps — sometimes AI handles a complex task better when it is decomposed.

I hit the free tier message limit

Wait until the limit resets (usually daily). In the meantime, use a different AI tool for basic tasks (ChatGPT, Gemini). If you consistently hit limits, consider the Pro plan — \$20/month is a reasonable professional tool investment.

AI gave me wrong information and I used it

This is why verification matters. Go back to the source and correct the error. In the future, verify all factual claims before publishing or sharing them. Treat AI as a draft, not a final source.

I feel like I am cheating by using AI

Review Day 6 of Week 1 on ethics. Using AI as a tool to enhance your work is not cheating — it is a skill. Using AI to misrepresent your abilities or submit someone else's work as your own is cheating. The line is disclosure and transformation: be transparent about your AI use, and always add your own expertise, judgment, and voice.

AI-generated images do not match my vision

Image prompting is a skill that takes practice. Be more specific about style, mood, composition, and details. Use the 'negative prompting' technique — specify what you do not want. Generate many variations and iterate on the best ones.

AI music generation sounds uncanny / not right

Current AI music generation is best for simple genres and short pieces. It struggles with complex forms, sophisticated harmony, and nuanced expression. Use it as a starting point or creative spark, not as a finished product.

I do not know which AI tool to use for a task

Ask Claude: 'I need to [describe task]. What is the best AI tool for this?' Claude can recommend tools even when it is not the best tool itself.

My website looks unprofessional

Start with a template rather than from scratch. Use consistent branding (colors, fonts) from your Week 4 brand guide. Ask Claude to review your site: paste a description of each page and ask for specific improvement suggestions.

I am overwhelmed by all the tools

You do not need all of them. Start with Claude for text tasks and Canva for visual tasks. Add other tools one at a time as specific needs arise. Your Week 5, Day 7 'Personal AI Toolkit' exercise helps you prioritize.

Appendix D — Glossary

This glossary covers every technical term used in this handbook and many you will encounter as you continue your AI journey.

Algorithm — A set of rules or instructions that a computer follows to solve a problem or complete a task. AI models are built on very complex algorithms.

API (Application Programming Interface) — A way for software programs to talk to each other. When an app uses Claude behind the scenes, it communicates through Claude's API.

Artificial Intelligence (AI) — Systems designed to perform tasks that typically require human intelligence — understanding language, recognizing patterns, making decisions.

Automation — Using technology to perform tasks without manual intervention. In the AI context, connecting tools together so actions happen automatically.

Bias — Systematic errors or prejudices in AI output, usually reflecting biases present in the training data or design decisions.

Chain-of-Thought — A prompting technique that asks AI to show its reasoning step by step, improving accuracy on complex tasks.

Chatbot — An AI system designed for conversation. Claude, ChatGPT, and Gemini are all chatbots, though the term undersells their capabilities.

Claude — An AI assistant made by Anthropic. The primary AI tool used in this course.

Content Strategy — A plan for what content to create, for whom, on what platforms, and how often.

Context Window — The amount of text an AI model can process at once — its working memory for the current conversation.

Copyright — Legal protection for original creative works. The application of copyright to AI-generated content is currently evolving.

CRM (Customer Relationship Management) — A system for tracking relationships with contacts, clients, or collaborators.

DALL-E — OpenAI's AI image generation model, integrated into ChatGPT.

Deep Learning — A type of machine learning using neural networks with many layers. The foundation of modern AI.

Diffusion Model — The AI architecture behind most image generators. Works by learning to reverse a process of adding noise to images.

EPK (Electronic Press Kit) — A digital collection of materials that represent a musician professionally — bio, photos, recordings, press quotes, and contact info.

Ethical AI — Approaches to developing and using AI that consider fairness, transparency, accountability, and societal impact.

Few-Shot Prompting — Providing examples in your prompt to show the AI what kind of output you want.

Fine-Tuning — Customizing a pre-trained AI model with additional specialized data.

Gemini — Google's AI model family, integrated with Google services.

Generative AI — AI that creates new content (text, images, audio, video) rather than just analyzing existing content.

GPT (Generative Pre-trained Transformer) — The model architecture behind ChatGPT. OpenAI regularly releases updated versions.

Hallucination — When AI generates confident-sounding but factually incorrect information.

Iteration — The process of refining AI output through successive prompts and adjustments.

Large Language Model (LLM) — An AI model trained on vast text data that can generate human-like text. Claude, GPT, and Gemini are all LLMs.

Machine Learning — A subset of AI where systems learn from data rather than being explicitly programmed.

Mastering — The final stage of audio production, optimizing a mix for distribution across different playback systems.

Metadata — Data about data — for music, this includes track titles, composer names, dates, and other organizational information.

Midjourney — An AI image generation tool known for artistic, high-quality output.

Model — In AI, a trained system that can perform tasks. 'Model' is used interchangeably with 'AI' in casual conversation.

Multimodal — AI that can process multiple types of input (text, images, audio, video) rather than just text.

Natural Language Processing (NLP) — The field of AI focused on understanding and generating human language.

Neural Network — A computing system inspired by the human brain, made up of interconnected nodes that process information.

No-Code — Tools that let you build digital products without writing programming code.

Open Source — Software whose code is publicly available and can be modified by anyone.

Parameter — An internal value in an AI model learned during training. More parameters generally means more capability.

Persona Stacking — A prompting technique that asks AI to respond from multiple perspectives simultaneously.

Prompt — Any input you give to an AI model. Crafting effective prompts is the core skill of AI fluency.

Prompt Engineering — The practice of writing prompts that produce optimal AI output.

RAG (Retrieval-Augmented Generation) — A technique where AI retrieves relevant information from a knowledge base before generating a response.

Responsible AI — Practices and principles for developing and deploying AI in ways that are safe, fair, and beneficial.

SEO (Search Engine Optimization) — Techniques for improving your website's visibility in search engine results.

Stem Separation — AI technology that isolates individual instruments or vocals from a mixed audio recording.

Structured Output — Requesting AI responses in specific formats (tables, lists, JSON) rather than free-form text.

Token — A unit of text (roughly a word or word part) that AI models use to process language.

Training — The process by which an AI model learns patterns from data before it is used by people.

Training Data — The data used to train an AI model. For language models, this is typically a large corpus of text.

Transformer — The neural network architecture behind modern LLMs. Introduced in 2017, it revolutionized AI.

Voice Cloning — AI technology that creates a synthetic voice matching a specific person's speech patterns.

Voice Synthesis — AI-generated speech from text input, also called text-to-speech (TTS).

Workflow — A sequence of steps (often involving multiple tools) to accomplish a task.

Zero-Shot Prompting — Asking AI to perform a task without providing examples — relying on its general training.

Appendix E — Recommended Resources & Further Learning

AI Courses (Free)

- Anthropic Academy (anthropic.skilljar.com) — Free courses with certificates, from beginner to advanced AI fluency
- Elements of AI (elementsofai.com) — University of Helsinki, 2M+ students, non-technical introduction
- Google AI Essentials — Through Grow with Google, covers AI fundamentals for professionals
- Andrew Ng's 'AI for Everyone' (Coursera) — Non-technical AI strategy and understanding
- Microsoft AI Skills (ai.azure.com) — Free AI learning paths for various skill levels

AI Newsletters

- The Neuron — Daily AI news digest, written for non-technical readers
- Ben's Bites — Daily AI news with a focus on practical applications
- Superhuman — AI news and tutorials for creative professionals
- AI for Musicians (Substack) — Niche newsletter focused on AI applications in music

Music Industry Resources

- New Music USA (newmusicusa.org) — Grants, opportunities, and resources for musicians
- Chamber Music America (chamber-music.org) — Commissioning grants and professional resources
- Fractured Atlas (fracturedatlas.org) — Fiscal sponsorship and arts management resources
- Future of Music Coalition (futureofmusic.org) — Research and advocacy on music industry issues

Books

- Co-Intelligence: Living and Working with AI by Ethan Mollick — The best book on practical AI fluency
- The Creative Act by Rick Rubin — Not about AI, but essential reading on creative process

- How Music Works by David Byrne — Useful context for understanding technology's relationship to music

Appendix F — Weekly Checklists & Progress Tracker

Use these checklists to track your progress through the course.

Week 1: AI Fundamentals & Prompt Mastery

- ☐ Day 1 lesson completed
- ☐ Day 1 exercises completed
- ☐ Day 1 journal entry written

- ☐ Day 2 lesson completed
- ☐ Day 2 exercises completed
- ☐ Day 2 journal entry written

- ☐ Day 3 lesson completed
- ☐ Day 3 exercises completed
- ☐ Day 3 journal entry written

- ☐ Day 4 lesson completed
- ☐ Day 4 exercises completed
- ☐ Day 4 journal entry written

- ☐ Day 5 lesson completed
- ☐ Day 5 exercises completed
- ☐ Day 5 journal entry written

- ☐ Day 6 lesson completed
- ☐ Day 6 exercises completed
- ☐ Day 6 journal entry written

- ☐ Day 7 lesson completed
- ☐ Day 7 exercises completed
- ☐ Day 7 journal entry written

Week 2: AI for Writing, Research & Communication

- ☐ Day 1 lesson completed
- ☐ Day 1 exercises completed
- ☐ Day 1 journal entry written

- ☐ Day 2 lesson completed
- ☐ Day 2 exercises completed
- ☐ Day 2 journal entry written

- ☐ Day 3 lesson completed
- ☐ Day 3 exercises completed
- ☐ Day 3 journal entry written

- ☐ Day 4 lesson completed
- ☐ Day 4 exercises completed
- ☐ Day 4 journal entry written

- ☐ Day 5 lesson completed
- ☐ Day 5 exercises completed
- ☐ Day 5 journal entry written

- ☐ Day 6 lesson completed
- ☐ Day 6 exercises completed

☐ Day 6 journal entry written

☐ Day 7 lesson completed

☐ Day 7 exercises completed

☐ Day 7 journal entry written

Week 3: AI Audio & Music Creation

☐ Day 1 lesson completed

☐ Day 1 exercises completed

☐ Day 1 journal entry written

☐ Day 2 lesson completed

☐ Day 2 exercises completed

☐ Day 2 journal entry written

☐ Day 3 lesson completed

☐ Day 3 exercises completed

☐ Day 3 journal entry written

☐ Day 4 lesson completed

☐ Day 4 exercises completed

☐ Day 4 journal entry written

☐ Day 5 lesson completed

☐ Day 5 exercises completed

☐ Day 5 journal entry written

☐ Day 6 lesson completed

- ☐ Day 6 exercises completed
- ☐ Day 6 journal entry written

- ☐ Day 7 lesson completed
- ☐ Day 7 exercises completed
- ☐ Day 7 journal entry written

Week 4: AI Visual & Video Creation

- ☐ Day 1 lesson completed
- ☐ Day 1 exercises completed
- ☐ Day 1 journal entry written

- ☐ Day 2 lesson completed
- ☐ Day 2 exercises completed
- ☐ Day 2 journal entry written

- ☐ Day 3 lesson completed
- ☐ Day 3 exercises completed
- ☐ Day 3 journal entry written

- ☐ Day 4 lesson completed
- ☐ Day 4 exercises completed
- ☐ Day 4 journal entry written

- ☐ Day 5 lesson completed
- ☐ Day 5 exercises completed
- ☐ Day 5 journal entry written

- ☐ Day 6 lesson completed
- ☐ Day 6 exercises completed
- ☐ Day 6 journal entry written

- ☐ Day 7 lesson completed
- ☐ Day 7 exercises completed
- ☐ Day 7 journal entry written

Week 5: AI Automation, Workflows & Business Tools

- ☐ Day 1 lesson completed
- ☐ Day 1 exercises completed
- ☐ Day 1 journal entry written

- ☐ Day 2 lesson completed
- ☐ Day 2 exercises completed
- ☐ Day 2 journal entry written

- ☐ Day 3 lesson completed
- ☐ Day 3 exercises completed
- ☐ Day 3 journal entry written

- ☐ Day 4 lesson completed
- ☐ Day 4 exercises completed
- ☐ Day 4 journal entry written

- ☐ Day 5 lesson completed
- ☐ Day 5 exercises completed
- ☐ Day 5 journal entry written

- ☐ Day 6 lesson completed
- ☐ Day 6 exercises completed
- ☐ Day 6 journal entry written

- ☐ Day 7 lesson completed
- ☐ Day 7 exercises completed
- ☐ Day 7 journal entry written

Week 6: Building Your Website with Claude Code

- ☐ Day 1 lesson completed
- ☐ Day 1 exercises completed
- ☐ Day 1 journal entry written

- ☐ Day 2 lesson completed
- ☐ Day 2 exercises completed
- ☐ Day 2 journal entry written

- ☐ Day 3 lesson completed
- ☐ Day 3 exercises completed
- ☐ Day 3 journal entry written

- ☐ Day 4 lesson completed
- ☐ Day 4 exercises completed
- ☐ Day 4 journal entry written

- ☐ Day 5 lesson completed
- ☐ Day 5 exercises completed

☐ Day 5 journal entry written

☐ Day 6 lesson completed

☐ Day 6 exercises completed

☐ Day 6 journal entry written

☐ Day 7 lesson completed

☐ Day 7 exercises completed

☐ Day 7 journal entry written

Week 7: AI for Your Music Career & Business

☐ Day 1 lesson completed

☐ Day 1 exercises completed

☐ Day 1 journal entry written

☐ Day 2 lesson completed

☐ Day 2 exercises completed

☐ Day 2 journal entry written

☐ Day 3 lesson completed

☐ Day 3 exercises completed

☐ Day 3 journal entry written

☐ Day 4 lesson completed

☐ Day 4 exercises completed

☐ Day 4 journal entry written

☐ Day 5 lesson completed

- ☐ Day 5 exercises completed
- ☐ Day 5 journal entry written

- ☐ Day 6 lesson completed
- ☐ Day 6 exercises completed
- ☐ Day 6 journal entry written

- ☐ Day 7 lesson completed
- ☐ Day 7 exercises completed
- ☐ Day 7 journal entry written

Week 8: How the Web Works & Your Development Environment

- ☐ Day 1 lesson completed
- ☐ Day 1 exercises completed
- ☐ Day 1 journal entry written

- ☐ Day 2 lesson completed
- ☐ Day 2 exercises completed
- ☐ Day 2 journal entry written

- ☐ Day 3 lesson completed
- ☐ Day 3 exercises completed
- ☐ Day 3 journal entry written

- ☐ Day 4 lesson completed
- ☐ Day 4 exercises completed
- ☐ Day 4 journal entry written

- ☐ Day 5 lesson completed
- ☐ Day 5 exercises completed
- ☐ Day 5 journal entry written

- ☐ Day 6 lesson completed
- ☐ Day 6 exercises completed
- ☐ Day 6 journal entry written

- ☐ Day 7 lesson completed
- ☐ Day 7 exercises completed
- ☐ Day 7 journal entry written

Week 9: Building Real Applications with Claude Code

- ☐ Day 1 lesson completed
- ☐ Day 1 exercises completed
- ☐ Day 1 journal entry written

- ☐ Day 2 lesson completed
- ☐ Day 2 exercises completed
- ☐ Day 2 journal entry written

- ☐ Day 3 lesson completed
- ☐ Day 3 exercises completed
- ☐ Day 3 journal entry written

- ☐ Day 4 lesson completed
- ☐ Day 4 exercises completed

☐ Day 4 journal entry written

☐ Day 5 lesson completed

☐ Day 5 exercises completed

☐ Day 5 journal entry written

☐ Day 6 lesson completed

☐ Day 6 exercises completed

☐ Day 6 journal entry written

☐ Day 7 lesson completed

☐ Day 7 exercises completed

☐ Day 7 journal entry written

Week 10: Deployment & Going Live

☐ Day 1 lesson completed

☐ Day 1 exercises completed

☐ Day 1 journal entry written

☐ Day 2 lesson completed

☐ Day 2 exercises completed

☐ Day 2 journal entry written

☐ Day 3 lesson completed

☐ Day 3 exercises completed

☐ Day 3 journal entry written

☐ Day 4 lesson completed

☐ Day 4 exercises completed

☐ Day 4 journal entry written

☐ Day 5 lesson completed

☐ Day 5 exercises completed

☐ Day 5 journal entry written

☐ Day 6 lesson completed

☐ Day 6 exercises completed

☐ Day 6 journal entry written

☐ Day 7 lesson completed

☐ Day 7 exercises completed

☐ Day 7 journal entry written

Week 11: Capstone Projects

☐ Capstone project chosen

☐ Project plan created

☐ Day 3 milestone completed

☐ Day 4 milestone completed

☐ Day 5 milestone completed

☐ Final polish and review

☐ Course reflection written

You do not need permission to start.

Open Claude. Begin.