# Vibe Coding Workshop Cheatsheet
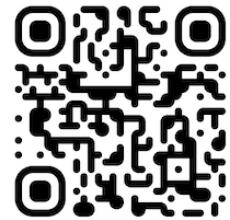
By Noah Eisenbruch - noaheisenbruch@gmail.com
Last updated: December 5, 2025
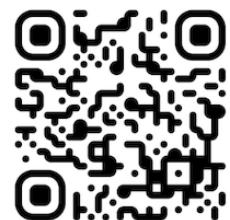
Access this cheatsheet, updates, resources, workshop projects, and more at:

https://eisenbruch.github.io/vibe-coding-workshops/

Click *Watch* on the GitHub repository to be notified of Cheatsheet updates and participant project uploads.

Your feedback is greatly appreciated and helps shape future workshops.

Please fill out our feedback form after the workshop: https://forms.gle/3iVRWgUS6o8U51Ut5

Table of Contents

Example Projects

Portfolio of Vibe-Coded Projects

1. Glitch QR (github.com/eisenbruch/glitch-qr)

- **Type:** QR code generation with a twist

- **Key Learning:** You can vibe code a simple tool in minutes, and it's fun

2. Instant Camera with Thermal Printer

- **Type:** Raspberry Pi with camera module, button, LED, 3D printed enclosure, and vibe coded software
- **Key Learning:** AI can assist with hardware prototyping and setup, not just software

3. Snark AI (snark-ai.com)

- **Type:** Satirical AI email service
- **Key Learning:** Vibe coding actually works to make services and websites

4. Nerktendo (nerktendo.com)

- **Type:** Retro interface and portfolio
- **Key Learning:** Sometimes AI is really bad at things. Try different tools and dive deeper into vibe coding.

5. LinkPee (linkp.ee)

- **Type:** Fully functional "link in bio" service with secure authentication & user profiles
- **Key Learning:** "Complex" features are absolutely achievable

6. Weather Worsener (nerktendo.com/weather-worsener)

- **Type:** Interactive game
- **Key Learning:** Fun and loose iterative projects are great for experimenting with concepts and mechanics and allow you to explore concepts and themes freely.

7. DomainFlip (nerktendo.com/domainflip)

- **Type:** Investor-ready startup demo
- **Key Learning:** Clear vision + detailed context = polished results

8. TrafficVision.Live (trafficvision.live)

- **Type:** Interactive map and database service with user focused features
- **Key Learning:** AI takes time to discover things like humans (optimization) and needs to take similar approaches a human would. They may need to be guided towards a solution.

> 💡 **Key Takeaway:** Don't limit yourself. If you think it should exist (no matter how silly), just try to create it. Vibe coding can make testing concepts easier than creating a mockup.

> 🧠 **Remember:** AI can help with more than just code - it can help prototype your hardware, organize your files, and much more.

Essential Vibe Coding Tools

Beginner: Browser-Based Tools

Best for first-time coders, quick prototypes, and learning the basics
No installation required - just open a browser and start

Chat Interfaces:

- **Claude.ai** - Advanced reasoning and coding
- **ChatGPT** - General purpose AI assistant
- **Gemini** - Google's multimodal AI

✅ **Pros:**

- No setup required
- Instant access from any device
- Great for learning and experimentation

❌ **Limitations:**

- Limited to simple projects
- Limited version control integration

> 💡 **Tip:** Start here to test ideas and learn prompting. When you're ready for real projects, move to intermediate or advanced tools.

Intermediate: Desktop Applications

Enhanced AI assistants with file system access
Best for document management, file organization, and basic coding tasks

Desktop Applications:

- **Claude Desktop** - Enhanced with file system access via MCP servers
- **ChatGPT Desktop** - Integrated workflow tools

✅ **Pros:**

- Full file system access
- Great for organizing files and documents
- MCP server support for extended capabilities
- Easy to use without coding experience

❌ **Limitations:**

- Not code-focused
- Slow file editing capabilities
- Not suited for complex development

> 💡 **Tip:** Desktop applications are excellent for organizing files and documents, and communicating with your computer in different ways beyond just coding.

Advanced: IDEs with AI Extensions & CLI (Command Line Interface) Tools

Professional development environments with powerful AI integration
Maximum power and flexibility for complex projects

AI Coding Tools:

All tools support full codebase understanding, git integration, MCP servers, and agents
Each available as both VS Code extension and standalone CLI

- **Claude Code** - Recommended for best overall quality and speed, especially for large projects
- **ChatGPT Codex** - Better at problem-solving with tricky bugs, but slower overall
- **Gemini Code Assist** - Recently updated with major improvements, great free option to get started

AI-Native IDEs:

- **Cursor**
- **Google Antigravity**

✅ **Pros:**

- Entire codebase context at once
- Powerful automation and agent capabilities
- Best for large refactoring and complex features
- Deep git integration
- Custom commands and workflows
- MCP server support for extended capabilities
- Professional development environment

❌ **Limitations:**

- More setup and configuration
- Steeper learning curve
- Requires understanding of general development practices
- Requires terminal comfort (for CLI tools)

> 💡 **Tip:** IDEs and CLI tools shine for large codebases and complex workflows. The investment in learning pays off quickly on real projects.

> 💡 **Tip:** Use Git version control! These tools make it easy to track changes, experiment safely, and collaborate professionally.

> 🧠 **Learn More:** See Working with AI Coding Agents for commands and advanced techniques.

> 🧠 **Learn More:** See the Extended AI Tools Directory for more specialized vibe coding platforms, web-hosted autonomous agents, and more.

Specialized Platforms

Browser-based IDEs with instant deployment and unique capabilities
Great for all skill levels - combines ease of use with professional features

- **Replit** - Collaborative coding environment with instant deployment, built-in database, and AI assistant
- **v0.dev** - Vercel's AI design tool that generates React components from text descriptions
- **Bolt.new** - StackBlitz's AI web development with full-stack support and instant preview
- **Google AI Studio** - Prototype with Gemini models, including code generation and multimodal AI

✅ **Pros:**

- No local setup required
- Instant deployment and hosting
- Share projects with a single link
- Built-in collaboration features
- Professional capabilities in a browser
- Great for rapid prototyping

❌ **Limitations:**

- Platform-specific workflows
- Less control over infrastructure
- May have usage limits on free tiers
- Dependent on platform's continued existence

> 💡 **Tip:** These platforms are perfect for quickly building and sharing projects without worrying about deployment. Use them alongside other tools depending on your needs.


Tool Selection Framework
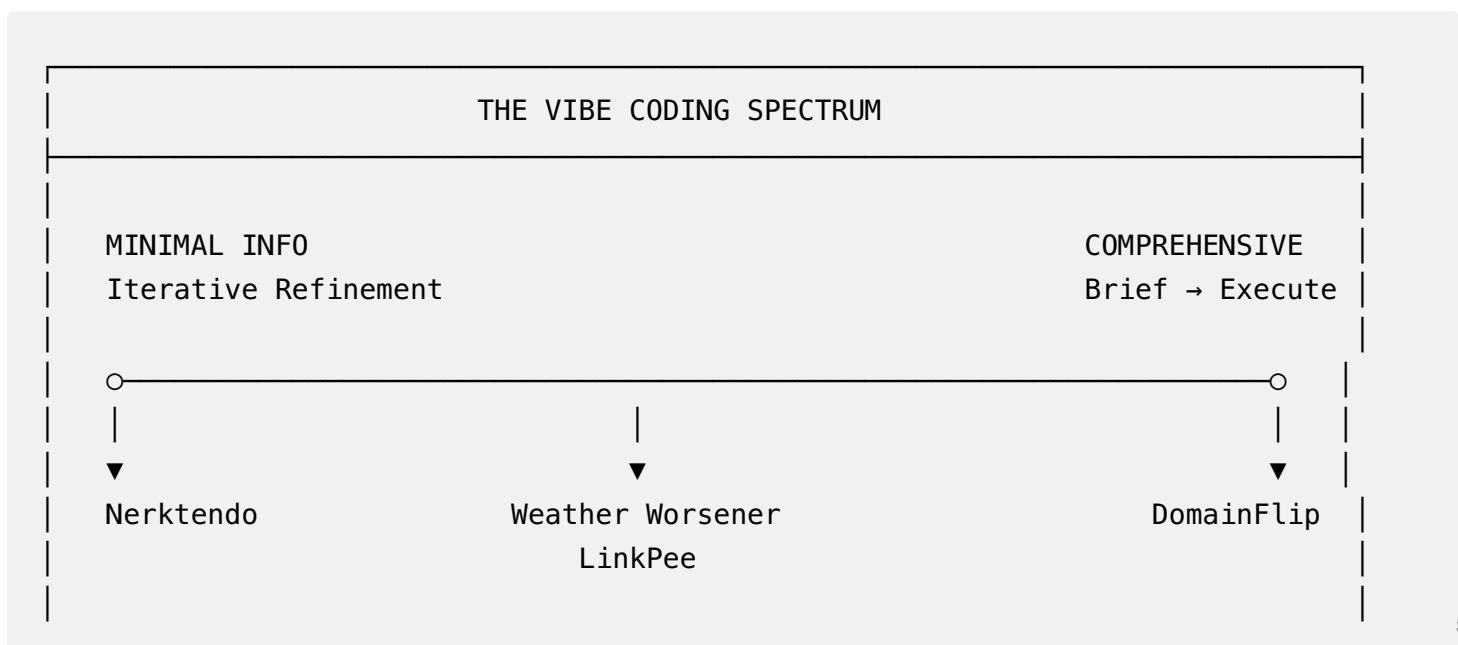
Ask yourself these questions:

1. **What are you building?** Simple tool/visual vs. complex app
2. **What's your experience level?** Never coded vs. comfortable with code
3. **What's your timeline?** Quick prototype vs. production-ready
4. **What's your budget?** Free tier vs. paid subscription
5. **Do you need automation?** Simple tasks vs. complex multi-step workflows

---

Core Approaches

The Vibe Coding Spectrum

All projects exist somewhere on this spectrum between two approaches:

```
┌─────────────────────────────────────────────────────────────────┐
│                    THE VIBE CODING SPECTRUM                       │
├───────────────────────────────────────────────────────────────────┤
│                                                                   │
│  MINIMAL INFO                                     COMPREHENSIVE   │
│  Iterative Refinement                             Brief → Execute │
│                                                                   │
│  o─────────────────────────────────────────────────────o         │
│  │                        │                        │    │         │
│  ▼                        ▼                        ▼    │         │
│  Nerktendo          Weather Worsener          DomainFlip │         │
│                        LinkPee                          │         │
│                                                         │         │
```

```
|   • Start broad              • Define everything |
|   • Refine through             upfront           |
|     conversation           • Features, style,    |
|   • Great for exploring        structure, rules  |
|   • Lower barrier           • Faster to polish    |
|                                                  |
└──────────────────────────────────────────────────┘
```

Approach 1: Minimal Info → Iterative Refinement

Example: Nerktendo

How it works: Start broad, refine through back-and-forth conversation

✅ **Pros:**

- Great for exploring possibilities and experimenting
- Sparks creative directions you wouldn't think of
- Lower barrier to starting

❌ **Cons:**

- Takes more back-and-forth
- May drift from original vision
- Can be difficult to guide refinement with minimal info

Approach 2: Comprehensive Brief → Execute

Example: DomainFlip

How it works: Define everything upfront - features, style, structure, functions, rules, edge cases

✅ **Pros:**

- Faster to a feature-full, polished result
- Stays true to vision
- Better for complex requirements

❌ **Cons:**

- Requires knowing what you want
- Less room for discovery
- Harder for beginners to write good briefs (but use AI to help)

How to do it: Use an LLM (not a coding AI) to create comprehensive planning documents before you start coding.

DomainFlip workflow:

1. **Planning phase**: I described my idea with all the details I had to an LLM chatbot

2. **Document creation**: The AI asked clarifying questions, then developed 8 detailed documents covering:
     ◦ Project overview and goals
     ◦ Technical architecture
     ◦ Design specifications
     ◦ Feature requirements and edge cases
3. **Execution phase**: I fed all documents to Claude Code in a single prompt
4. **Result**: One hour later, the complete site was built exactly to spec

The key: comprehensive upfront documentation eliminates ambiguity and reduces iteration.

> 💡 **Pro Tip:** Use an AI chatbot (not coding assistant) to plan your project in detail, then feed that plan to an AI coding agent. DomainFlip was created with only 1 Claude Code prompt, but it was fed 8 different multi page documents outlining the entire project in extreme detail.

> 🎁 **Bonus:** Read through the DomainFlip summary and build documents

---

Good Prompts vs Bad Prompts

❌ Bad Prompt	✅ Good Prompt
"Make me a website"	"Create a single-page portfolio site with a hero section, 3 project cards, and a contact form. Use a dark theme with blue accents."
"Fix it"	"The submit button doesn't work. When I click it, nothing happens. No errors in console. Here's the relevant code: [paste code]"
"Add a database"	"Add a SQLite database to store user tasks. Include fields for: id, title, description, due_date, completed (boolean). Create functions for CRUD operations."
"Make it look better"	"Improve the card component: add subtle shadows, rounded corners (8px), and a hover effect that slightly lifts the card."
"It's broken"	"Expected: clicking 'Save' should show a success message. Actual: page refreshes and data is lost. Browser: Chrome. Error in console: [paste error]"

Key principles:

- Be specific about what you want
- Include relevant context (tech stack, constraints, goals)
- When reporting bugs: expected vs actual behavior + any errors
- Describe the outcome, not just the task

Working with AI Coding Agents

This section focuses on Claude Code but the concepts and techniques apply to all agents.

Getting Started

Requirement: Claude Pro/Max subscription or Enterprise access (API credits supported but not recommended)

Option 1: VS Code Extension

1. Open VS Code
2. Go to Extensions (Ctrl+Shift+X / Cmd+Shift+X)
3. Search for "Claude Code"
4. Click Install
5. Authenticate with your Claude account
6. The Claude icon will now show up in your window. Click to get started.

Option 2: CLI Installation

Choose the installation method for your system:

macOS/Linux (Homebrew):

```
brew install --cask claude-code
```

Note: Requires Homebrew to be installed first

macOS/Linux/WSL:

```
curl -fsSL https://claude.ai/install.sh | bash
```

Windows (PowerShell):

```
irm https://claude.ai/install.ps1 | iex
```

Windows (CMD):

```
curl -fsSL https://claude.ai/install.cmd -o install.cmd && install.cmd && del instal
```

Once installed, navigate to your project directory and run `claude` to start.

> 🧠 **Learn More:** Read Anthropic's official Claude Code Best Practices guide for workflows, optimization tips, and advanced techniques.

Claude Code CLI Commands

Command	Purpose
claude	Start Claude in your terminal.
esc	Interrupt Claude while it's working.
esc esc	Go back in conversation and code history.
/init	Review codebase and create CLAUDE.md documentation file for AI reference
#memory	Easily add to CLAUDE.md memory
/compact	Compact conversation to clear space while keeping some memory
/clear	Better space clearing but loses memory (or quit/restart Claude Code)
/usage	Check current usage level
/resume	Resume a previous conversation
/review	Code review

> 💡 **Tip:** Store important instructions in CLAUDE.md to prevent AI from doing unwanted things (overwriting data, unnecessary builds, etc.). CLAUDE.md files exist for your user and each individual project folder where you use Claude Code. Use this to your advantage - global knowledge vs project specific knowledge.

Vibe Coding Essential Techniques

Interrupt When Needed

Sometimes AI goes off the rails. If you see it going down the wrong path, interrupt it, go back (double tap esc), and give better instructions. This saves memory and credits vs. correcting mistakes later. AI is bad at remembering what it did wrong.

Self-Checking

Ask your AI to check its own (or your) work for bugs and see if there's room for improvement. Optimizing for resources and speed is generally good (unless it isn't).

Working Through Stubborn Bugs

When you have a bug AI can't solve after multiple attempts:

1. Use Claude thinking levels. Type: think, think hard, think harder, or ultrathink
2. Try a different coding agent (e.g. ChatGPT Codex)
3. Ask 2 different AIs the same question
   1. Give each AI the other's response for analysis
   2. Have both provide updated plans
   3. Give both responses to a new agent for comprehensive summary
   4. Have another new agent read and implement the solution

Debugging Workflow

AI agents fix issues best when given all relevant information:

1. Use browser DevTool Console to look for errors and warnings: AI agents love reading errors and warnings
2. Copy and paste HTML from the DevTool Element Inspector if unsure how to describe an issue
3. Describe what you want and what the issue is in detail
4. Provide full context especially if debugging in a fresh conversation

---

Claude Code Advanced Features

Claude Agents (Tasks)

Multiple Claude Code instances using skills and tools to autonomously complete complex workflows in parallel.

Use case: Analyze reports or data sources simultaneously

How to use: Just ask Claude to start some agents (tasks)

Custom Commands

Shortcuts for common workflows without detailed descriptions each time.

Use case: Super helpful for scraping and cleaning data

Examples: `/new-source` and `/commit` (TrafficVision.Live)

Git Worktrees

Git worktrees let you work on multiple branches simultaneously without switching contexts or creating separate repository clones.

What it solves: Working on a feature while needing to quickly fix a bug on main, or testing different approaches side-by-side

How to use:

1. Create a new worktree:

```
git worktree add ../project-feature-branch feature-branch-name
```

1. Claude Code can work across worktrees simultaneously
2. Each worktree maintains its own working directory and branch
3. Shared git history means commits from any worktree are available everywhere

Use cases:

- Developing a new feature while maintaining production bug fixes
- Testing multiple implementation approaches in parallel

- Code review without disrupting your current work
- Running different versions for comparison testing

Clean up when done:

```
git worktree remove ../project-feature-branch
```

> 💡 **Tip:** Name worktree directories clearly (e.g., `project-name-feature-name`) to avoid confusion when multiple Claude Code instances are running.

> 💡 **Tip:** If worktrees are confusing to you (or if any advanced workflow is), ask Claude to do it for you and teach you along the way.

MCP Servers

Model Context Protocol (MCP) servers extend Claude's capabilities by connecting to external tools, APIs, databases, and services through a standardized protocol.

What it is: An open standard that lets AI assistants securely connect to external data sources and tools without custom integrations for each service.

Why use it:

- Access real-time data beyond Claude's training cutoff
- Connect to your tools (GitHub, Slack, databases, file systems)
- Automate workflows across multiple services
- Get specialized functionality (web search, documentation, browser control)

Common MCP Servers:

- **Filesystem** - Read/write local files with permission-based access
- **GitHub** - Manage PRs, issues, and repository operations
- **Puppeteer** - Browser automation and web scraping
- **Context7** - Up-to-date library documentation with version-specific references
- **Brave Search** - Real-time web search results
- **Sequential Thinking** - Break down complex tasks into structured steps
- **Google Drive** - Access and manage Drive documents
- **Slack** - Send messages and interact with Slack workspaces

Installation - Claude Desktop:

1. Install Node.js if needed: `brew install node` (macOS)
2. Navigate to Settings → Extensions in Claude Desktop
3. Click "Browse extensions" to install from directory (one-click)

Installation - Claude Code:

```
# Add server with user scope (available in all projects)
claude mcp add github --scope user

# Add with environment variables
```

```
claude mcp add -e API_KEY=your_key servername -- npx @org/server

# List configured servers
claude mcp list

# Check server status
/mcp
```

Configuration Scopes:

- **Local** (default) - Only available in current directory
- **User** (`--scope user`) - Available globally across all projects
- **Project** (`.mcp.json`) - Shared with team via version control

> 💡 **Tip:** Use user scope for personal tools, project scope for team-shared servers. After adding servers, restart Claude Desktop or Claude Code to activate them.

Config File Locations:

- **macOS:** `~/Library/Application Support/Claude/claude_desktop_config.json`
- **Windows:** `%APPDATA%\Claude\claude_desktop_config.json`
- **Claude Code:** `~/.claude.json` (user scope) or `.mcp.json` (project scope)

Finding More Servers:

- Official servers: github.com/modelcontextprotocol/servers
- Community directory: mcp.so
- Smithery: smithery.ai

Beyond Coding

Claude and other AI agents can help with more than just code if they have access to the right tools:

- File and document organization
- Obsidian integration through filesystem
- Note analysis and knowledge management
- Similar capabilities to IDE integration

Project Ideas

Customize these to your liking. Extended prompts with full details available on the GitHub repo.

Simple Projects

To-Do List

Task management with add/edit/delete, checkboxes, filters (all/active/completed), localStorage.

```
Create a to-do list app with add/edit/delete tasks, checkboxes, filter by all/active,
```

Habit Tracker

Daily habits with check-offs, 7-day calendar view, streak counter.

```
Build a habit tracker: add custom habits, mark complete daily, 7-day streak calendar
```

Calculator

Basic math (+, -, ×, ÷), keyboard support, iOS or retro style.

```
Create a calculator with basic operations, clear/backspace, keyboard support. Style
```

Guessing Game

Number guessing 1-100, higher/lower hints, attempt counter, difficulty levels.

```
Build a guessing game: random number 1-100, higher/lower hints, track attempts, play
```

Medium Projects

Weather App

City search, 5-day forecast, geolocation, weather-themed gradients. **API:** OpenWeatherMap

```
Build a weather app: city search, 5-day forecast, weather icons, humidity, wind, "fe
```

Portfolio Site

Hero section, project cards with filtering, contact form, dark/light mode. **API:** EmailJS

```
Build a portfolio: animated hero, project showcase with category filters, skills sec
```

Pomodoro Timer

25/5 min sessions, customizable durations, audio alerts, session stats.

```
Build a pomodoro timer: 25-min work / 5-min break sessions, customizable durations, a
```

Recipe Finder

Search by ingredient, dietary filters, save favorites, random recipe. **API:** Spoonacular

```
Build a recipe app: search by ingredient/dish, filter by dietary restrictions, show
```

Complex Projects

Real-Time Chat

Discord/Slack clone with auth, channels, typing indicators, read receipts, file uploads.

```
Build a real-time chat app: user auth, multiple channels, online/offline status, typ
```

Restaurant Reservations

OpenTable clone with restaurant browsing, table booking, calendar availability, admin panel.

```
Build a reservation system: browse restaurants with photos/menus, search/filter, inte
```

Data Dashboard

Analytics visualizer with multiple chart types, date filters, CSV upload, drag-drop layout.

```
Build an analytics dashboard: line/bar/pie charts + heatmaps, interactive filtering,
```

Shift Scheduler

Staff scheduling with availability, auto-scheduling algorithm, fairness tracking.

```
Build a shift scheduler: employee profiles with qualifications + availability, auto-s
```

Bonus Challenge

Open Source Contribution - Find a GitHub project you use, pick an issue or improvement, and submit a PR. Great for learning real codebases and professional workflows.

Find projects: GitHub Explore, First Timers Only, Good First Issue, Up For Grabs

Submitting Your Workshop Project

Share your work! The workshop repo is a living archive of our projects.

View submission instructions at: **eisenbruch.github.io/vibe-coding-workshops**

Your AI assistant can guide you through the Git/GitHub process.

Extended AI Tools Directory

> ⚠️ **Note:** Projects shut down and get acquired frequently. Some tools may no longer exist.

Vibe Coding & Development Tools

Claude Code - claude.ai/code
Terminal-based AI coding assistant with full codebase context, git integration, and MCP server support

Gemini Code Assist - codeassist.google
Google's AI coding assistant available as IDE extension and CLI, recently updated with major improvements

Google Antigravity - antigravity.google
Agentic development platform with autonomous agents and Manager View for coordinating AI workflows

ChatGPT Codex - chatgpt.com/codex
OpenAI's code-specialized model integrated into ChatGPT for code generation and understanding

Cursor - cursor.com
AI-native code editor built on VS Code with chat, inline editing, and codebase understanding

GitHub Copilot - github.com/features/copilot
AI pair programmer integrated into VS Code, JetBrains IDEs, and other editors

Google AI Studio - aistudio.google.com
Google's platform for prototyping with Gemini models, including code generation

v0.dev - v0.dev
Vercel's AI tool for generating React components and UI designs from text descriptions

Bolt.new - bolt.new
StackBlitz's AI web development platform that builds and deploys full-stack apps instantly

Windsurf - codeium.com/windsurf
AI-powered IDE with Cascade flows for multi-file editing and autonomous task completion

Supermaven - supermaven.com
Fast AI code completion with 1 million token context window

Aider - aider.chat
Command-line AI coding assistant that edits code in your local git repository

Replit - replit.com
Collaborative browser-based IDE with AI assistance, instant deployment, and multiplayer coding

Image & Video

Google Nano Banana - aistudio.google.com
One of the newest and best image generation and editing models.

Google Veo 3 - deepmind.google
Advanced video generation with 8-second clips and synchronized native audio

Google Flow - labs.google/fx/tools/flow
Google's experimental video generation tool from Google Labs FX

Kling AI - klingai.com
Versatile image and video platform with reference-based generation, video restyling, next shot generation, and start/end frame controls

Runway - runwayml.com
Advanced AI video platform with Gen-4 models for text/image-to-video

Midjourney - midjourney.com
Discord-based platform creating high-quality artistic images from text prompts

DALL-E - openai.com/dall-e
OpenAI's image generator for detailed images from natural language

Pika - pika.art
AI video generator creating cinematic clips up to 10 seconds

Decohere - decohere.ai
Generates characters, images, and videos from text and image prompts

Stability AI - stability.ai
Creator of Stable Diffusion for multimodal generative AI

Leonardo.ai - leonardo.ai
Fine-tuned models for game assets, concept art, and production visuals

Hedra - hedra.com
Specializes in talking avatar generation with realistic lip-sync

HeyGen - heygen.com
AI avatar platform for professional spokesperson videos

Synthesia - synthesia.io
AI presenters for training videos and marketing content

Adobe Firefly Video - firefly.adobe.com
Text/image-to-video with 1080p output and commercially safe generation

Hailuo AI - hailuoai.video
Chinese AI video generator creating 5-6 second HD videos

Higgsfield - higgsfield.ai
Video and image generation with a focus on impossible effects and transitions

Fooocus - GitHub
Open-source Stable Diffusion interface with advanced inpainting

AUTOMATIC1111 - GitHub
Popular web interface for Stable Diffusion with extensive features

Switchlight - switchlight.beeble.ai
AI-powered video relighting tool with PBR maps

Adobe Express Image Background Remover - adobe.com
Free AI tool for instant background removal with transparent PNG output

Adobe Express Video Background Remover - adobe.com
Remove video backgrounds with transparent MP4 output

VideoBGRemover - videobgremover.com
Professional background removal for videos up to 3 hours

Audio & Music

Hume - hume.com
Text-to-speech and speech-to-speech editor with voice cloning and developer APIS for analyzing emotional expression and generating expressive speech.

Descript - descript.com
All-in-one editor with text to speech, voice cloning, AI transcription and text-based editing

ElevenLabs - elevenlabs.io
Advanced AI voice cloning and text-to-speech in 30+ languages

Suno - suno.ai
AI music generation with vocals and instruments from text

Udio - udio.com
Professional-quality music tracks from text descriptions

Adobe Podcast Enhance - podcast.adobe.com
AI audio enhancement removing background noise

AI Chat & Search

Perplexity - perplexity.ai
AI-powered search with sourced answers and citations

DeepSeek - deepseek.com
Chinese open-source AI with powerful reasoning (R1 model)

Venice.ai - venice.ai
Privacy-focused uncensored AI with local data storage

Maple.ai - trymaple.ai
Privacy-focused with end-to-end encryption and confidential computing

Z.ai Chat - chat.z.ai
Free AI chatbot with presentation generation and multimodal capabilities

AI Infrastructure & Development

Replicate - replicate.com
Cloud platform for running open-source AI models via API

Hugging Face - huggingface.co
The GitHub of AI - hub for models, datasets, and applications

Groq - groq.com
Ultra-fast AI inference with custom LPU chips

Chutes.ai - chutes.ai
Decentralized serverless AI compute platform

3D & Spatial

Depth Anything V2 - GitHub
Monocular depth estimation for 3D reconstruction

Luma Labs - lumalabs.ai
Interactive 3D scenes using NeRF and Gaussian Splatting

Hyper3D (Rodin AI) - hyper3d.ai
Production-ready 3D models from text or images

Tripo3D - tripo3d.ai
Fast 3D asset generation in ~10 seconds

TRELLIS - Hugging Face
Microsoft's 3D generation with Radiance Fields and meshes

HitPaw 3D (Hitem3D) - hitem3d.ai
Ultra-high resolution 3D models from single images

Marble (World Labs) - marble.worldlabs.ai
Generate 3DGS worlds from text, images, or video.

Brush - GitHub
Open-source 3D reconstruction with Gaussian Splatting

COLMAP - colmap.github.io
Open-source photogrammetry pipeline for 3D reconstruction

RealityScan - realityscan.com
Professional photogrammetry by Epic Games

Postshot - jawset.com
Desktop app for training NeRF and Gaussian Splatting locally

Agentic Browsers

AI-native browsers that autonomously navigate websites, complete tasks, and execute workflows.

BrowserOS - github.com/browseros-ai/BrowserOS
BrowserOS is an open-source chromium fork that runs AI agents natively. Your open-source, privacy-first alternative to ChatGPT Atlas, Perplexity Comet, Dia.

ChatGPT Atlas - chatgpt.com/atlas
ChatGPT in a browser with memory and automation abilities

Perplexity Comet - perplexity.ai/comet
AI-first browser with autonomous task execution and integrated search

Opera Neon - operaneon.com
Premium agentic browser with Chat, Do, and Make functions

Fellou - fellou.ai
Self-driving browser with deep search and agentic memory

Sigma AI Browser - sigmabrowser.com
Privacy-first with end-to-end encryption and Deep Research mode

Dia Browser - diabrowser.com
Tab-aware assistant with Skills system and natural language commands

---

Specialized Tools

ComfyUI - comfy.org
Node-based workflow platform for multiple AI model types

Genspark - genspark.ai
Many AI tools in one chatbot

Cascdr - cascdr.xyz
Bitcoin Lightning-powered AI workflow platform

Browserbase - browserbase.com
Developer infrastructure for headless browsers and AI agents

Manus.im - manus.im
Browser-based AI agent platform for autonomous tasks

n8n - n8n.io
Flexible AI workflow automation with visual node-based editor and 400+ integrations

---

Resources & Support

For more resources, tutorials, and support, visit the workshop materials repository or reach out.

https://github.com/eisenbruch/vibe-coding-workshops
noaheisenbruch@gmail.com

Cascdr - cascdr.xyz
Bitcoin Lightning-powered AI workflow platform

Browserbase - browserbase.com
Developer infrastructure for headless browsers and AI agents

Manus.im - manus.im
Browser-based AI agent platform for autonomous tasks

n8n - n8n.io
Flexible AI workflow automation with visual node-based editor and 400+ integrations