

# Vibe Coding Workshop Cheatsheet

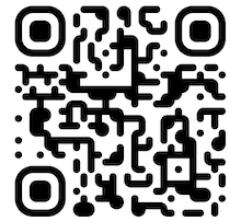
---

By Noah Eisenbruch - [noah.eisenbruch@gmail.com](mailto:noah.eisenbruch@gmail.com)

Last updated: November 25, 2025

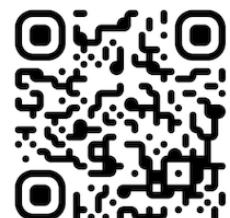
Access this cheatsheet, updates, resources, workshop projects, and more at:  
<https://eisenbruch.github.io/vibe-coding-workshops/>

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

1. Example Projects
  2. Vibe Coding Tools
  3. Core Approaches
  4. Working with Claude Code
  5. Project Ideas
  6. Submitting Your Workshop Project
  7. Extended AI Tools Directory
- 

## Example Projects

### Portfolio of Vibe-Coded Projects

#### 1. Glitch QR ([github.com/eisenbruch/glitch-qr](https://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. Snark AI ([snark-ai.com](https://snark-ai.com))

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

#### 3. Nerkendo ([nerktendo.com](https://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.

#### 4. LinkPee ([linkp.ee](https://linkp.ee))

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

#### 5. Weather Worsener ([nerktendo.com/weather-worsener](https://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.

#### 6. DomainFlip ([nerktendo.com/domainflip](https://nerktendo.com/domainflip))

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

#### 7. TrafficVision.Live ([trafficvision.live](https://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.

#### 8. 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

 **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

### Choose Your Level

#### 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 Claude Code](#) for commands and advanced techniques.

 **Learn More:** See the [Extended AI Tools Directory](#) for more specialized AI 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:

#### **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.

 **Bonus:** Read through the [DomainFlip summary and build documents](#)

## Finding Your Approach

Most projects fall somewhere in the middle. Weather Worsener and LinkPee are examples of balanced approaches.

 **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.

Most of my examples are website related

## Working with Claude Code

### 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
- 

## 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 install.cmd
```

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
<code>claude</code>	Start Claude in your terminal.
<code>esc</code> <code>esc</code>	Go back in conversation and code history.
<code>/init</code>	Review codebase and create <a href="#">CLAUDE.md</a> documentation file for AI reference
<code>#memory</code>	Easily add to <a href="#">CLAUDE.md</a> memory
<code>/compact</code>	Compact conversation to clear space while keeping some memory

Command	Purpose
/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.

## 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: think, think hard, think harder, 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

# 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_keyservername -- 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/clause\_desktop\_config.json
- **Windows:** %APPDATA%\Claude\clause\_desktop\_config.json
- **Claude Code:** ~/.clause.json (user scope) or .mcp.json (project scope)

## Finding More Servers:

- Official servers: [github.com/modelcontextprotocol/servers](https://github.com/modelcontextprotocol/servers)
- Community directory: [mcp.so](https://mcp.so)
- Smithery: [smithery.ai](https://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

### Simple Projects

*These starter prompts are to give you ideas. Change them to your liking.*

#### To-Do List

##### Starter Prompt:

Create a to-do list app with the following features:

- Add, edit, and delete tasks
- Mark tasks as complete with a checkbox
- Filter view by "all", "active", and "completed"
- Tasks persist in localStorage

Use vanilla JavaScript or React, make it clean and minimal with a soft color palette

## Habit Tracker

### Starter Prompt:

Build a habit tracker that lets users:

- Add custom habits to track
- Mark habits as complete for each day
- Show a 7-day streak calendar view
- Calculate and display current streak

Use a grid layout with green checkmarks for completed days. Make it motivating and visually appealing.

## Custom Calculator

### Starter Prompt:

Create a calculator with:

- Basic operations (+, -, ×, ÷)
- Clear and backspace functions
- Keyboard support for number entry
- Display for current input and result

Style it like a modern iOS calculator or create a retro/vintage aesthetic. Add smooth transitions and responsive design.

## Basic Game

### Starter Prompt:

Build a simple guessing game where:

- Computer picks a random number between 1-100
- Player gets hints ("higher" or "lower") after each guess
- Track number of attempts
- Show win message and option to play again

Make it colorful and fun with emoji reactions to guesses. Add a difficulty selector for different ranges.

## Medium Projects

### Weather App

### Starter Prompt:

Build a weather app that includes:

- Feature 1: Search for any city and display current weather
- Feature 2: 5-day forecast with high/low temperatures
- Feature 3: Display weather icons, humidity, wind speed, and "feels like" temp
- Feature 4: Geolocation to auto-detect user's location

Design should be clean and modern with weather-appropriate background gradients (sunrise/sunset).

Use OpenWeatherMap API or WeatherAPI.

Target audience: anyone who wants quick weather info at a glance

## Portfolio Website

### Starter Prompt:

Build a personal portfolio website that includes:

- Feature 1: Hero section with animated introduction and call-to-action
- Feature 2: Project showcase with filtering by technology/category
- Feature 3: About section with skills visualization (progress bars or icons)
- Feature 4: Contact form with email integration (EmailJS or similar)
- Feature 5: Dark/light mode toggle

Design should be modern and professional with smooth scroll animations and transitions.

Use React or Next.js with Tailwind CSS.

Target audience: potential employers and clients

## Pomodoro Timer App

### Starter Prompt:

Build a productivity timer app that includes:

- Feature 1: 25-minute work sessions with 5-minute breaks
- Feature 2: Customizable timer durations
- Feature 3: Audio notification when timer completes
- Feature 4: Session counter and daily statistics
- Feature 5: Pause, resume, and reset controls

Design should be minimalist and distraction-free with a large, readable timer display.

Add ambient background sounds (optional toggle).

Target audience: students and remote workers seeking better focus

## Recipe Finder

### Starter Prompt:

Build a recipe search app that includes:

- Feature 1: Search recipes by ingredient or dish name
- Feature 2: Filter by dietary restrictions (vegetarian, vegan, gluten-free)
- Feature 3: Display recipe details including ingredients, instructions, and cook time
- Feature 4: Save favorite recipes to localStorage
- Feature 5: Random recipe generator for inspiration

Design should be warm and inviting with food photography and card-based layouts.

Use Spoonacular API or Edamam API.

Target audience: home cooks looking for meal ideas

# Complex Projects

## Real-Time Chat App

### Starter Prompt:

Build a chat application where users can message each other in real-time with these

#### Core Features:

- User accounts with secure registration and login
- Instant messaging that updates without refreshing the page
- Multiple chat rooms or channels for different conversations
- See who's online/offline right now
- "User is typing..." indicators
- Message history that loads older messages as you scroll up
- Timestamps showing when each message was sent
- Read receipts (checkmarks when messages are seen)
- Emoji picker for reactions
- Ability to upload and share images/files

#### User Experience:

- Clean, modern interface similar to Discord or Slack
- Sidebar showing all available channels
- Main chat area for conversation
- Works smoothly on both desktop and mobile
- Fast and responsive – messages appear instantly

#### Technical Requirements:

- Secure authentication and password storage
- Protect against common security issues
- Store all messages and user data in a database
- Deploy the app so others can access it online

Target audience: Teams, friend groups, or communities who want their own private chat

## Restaurant Reservation System

### Starter Prompt:

Build a restaurant reservation platform where customers can book tables and restaurants

#### Customer Features:

- Browse restaurants with photos, menus, and descriptions
- Search and filter by cuisine type, location, and price range
- Interactive calendar showing available reservation times
- See real-time availability (which time slots are open)
- Book a table by selecting date, time, party size, and adding special requests

- Receive email confirmation immediately after booking
- Get SMS reminder 24 hours before the reservation
- Personal dashboard to view, modify, or cancel upcoming reservations
- Works perfectly on mobile phones for booking on-the-go

#### Restaurant Owner Features:

- Admin panel to view all bookings for their restaurant
- Manage table capacity and available time slots
- See booking details (party size, special requests, contact info)
- Prevent double-booking automatically
- Update restaurant information, photos, and menu

#### User Experience:

- Elegant, trustworthy design similar to OpenTable or Resy
- High-quality imagery that makes restaurants look appealing
- Fast loading and smooth interactions
- Handle different time zones correctly

#### Technical Requirements:

- Secure login for both customers and restaurant owners
- Store all restaurant data, bookings, and user accounts safely
- Send automated emails and SMS notifications
- Deploy online so anyone can use it

Target audience: Diners looking for an easy way to book tables, and restaurant owners

## Interactive Data Visualization Dashboard

### Starter Prompt:

Build an analytics dashboard that turns data into beautiful, interactive visualizations.

#### Core Features:

- Display data in multiple chart types: line graphs, bar charts, pie charts, and heatmaps
- Click and interact with charts to explore details
- Filter data by date ranges (today, last week, last month, custom range)
- Watch data update live in real-time as new information comes in
- Upload data from CSV or JSON files
- Export visualizations and data to PDF or CSV
- Customize the dashboard layout by dragging and dropping different widgets
- Works smoothly on tablets and desktop computers

#### Data Management:

- Store historical data so you can analyze trends over time
- Handle large datasets without slowing down
- Automatically calculate key metrics and trends
- Refresh data on a schedule (hourly, daily, etc.)

#### User Roles:

- Viewers: can only see dashboards
- Editors: can modify dashboards and settings
- Admins: full control over users and data

#### User Experience:

- Professional interface similar to Google Analytics or Tableau
- Clear visual hierarchy showing what's important
- Intuitive controls that are easy to understand
- Fast performance even with lots of data
- Helpful loading states and error messages

#### Technical Requirements:

- Secure user authentication with different permission levels
- Store data in a reliable database
- Deploy online for team access

#### Example use cases:

- Sales dashboard showing revenue, top products, and customer trends
- Social media analytics tracking engagement and follower growth
- IoT sensor monitoring for temperature, humidity, or equipment status

Target audience: Business teams, data analysts, or anyone who needs to understand the

## Employee Shift Scheduler

#### Starter Prompt:

Build a smart shift scheduling system that creates fair schedules automatically while

#### Staff Management:

- Add employees with their qualifications (certifications, skills, roles)
- Set each person's availability for every day of the week and their shift preferences
- View each employee's upcoming schedule and total hours

#### Automatic Scheduling:

- Generate schedules automatically that:
  - Only assign people when they're available
  - Match qualifications to shift requirements
  - Balance workload fairly across multiple weeks (not just one week at a time)
  - Avoid scheduling someone for multiple shifts on the same day
  - Respect people's preferences when possible
  - Track historical fairness so everyone gets their preferred shifts over time

#### Shift Templates:

- Create recurring weekly shift patterns (morning, afternoon, night, weekend)

- Define how many people are needed for each shift
- Specify what qualifications are required
- Handle shifts that span multiple days

#### Schedule Management:

- View schedules across multiple weeks at once
- Manually adjust assignments if needed (drag and drop)
- Clear entire weeks and regenerate schedules
- Remove individual assignments and fill them automatically
- See visual warnings for conflicts or issues

#### Fairness Dashboard:

- Show how fairly preferences have been fulfilled across past weeks
- Look ahead to see projected fairness for future schedules
- Adjust the time window (1-12 weeks) to analyze different periods
- Visual indicators for workload balance and preference scores

#### User Experience:

- Clean, professional interface focused on clarity
- Color coding for different shift types
- Visual indicators showing workload balance
- Easy to spot scheduling conflicts
- Export schedules to PDF or CSV for printing/sharing
- Works on desktop and tablet

#### Technical Requirements:

- Store all staff data, shifts, and schedules in a database
- Smart algorithm that creates fair, conflict-free schedules
- Fast performance even with many employees and weeks of data

Target audience: Small business managers, healthcare facilities, retail stores, or a

---

## Bonus Challenge

### **Open Source Contribution Challenge**

Take a project that you use from GitHub, come up with an improvement, and submit a PR (pull request).

#### **How to get started:**

1. Find a project on GitHub that is interesting or useful to you
2. Set up the project locally and explore the codebase
3. Pick an existing issue or come up with an improvement (bug fix, feature, documentation, tests)
4. Fork the repo, create a branch, and make your changes
5. Write clear commit messages and test your changes
6. Submit a pull request with a detailed description

## This teaches you about:

- Reading and understanding existing codebases
- Contributing to open source communities
- Professional collaboration workflows
- Git and version control best practices
- Code review process and feedback incorporation

## Recommended platforms to find projects:

- GitHub Explore ([github.com/explore](https://github.com/explore))
- First Timers Only ([firsttimersonly.com](https://firsttimersonly.com))
- Good First Issue ([goodfirstissue.dev](https://goodfirstissue.dev))
- Up For Grabs ([up-for-grabs.net](https://up-for-grabs.net))

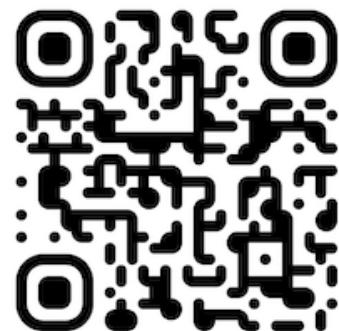
---

## 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](https://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](https://claude.ai/code)

Terminal-based AI coding assistant with full codebase context, git integration, and MCP server support

#### **Gemini Code Assist** - [codeassist.google](https://codeassist.google)

Google's AI coding assistant available as IDE extension and CLI, recently updated with major improvements

#### **Google Antigravity** - [antigravity.google](https://antigravity.google)

Agentic development platform with autonomous agents and Manager View for coordinating AI workflows

#### **Google AI Studio** - [aistudio.google.com](https://aistudio.google.com)

Google's platform for prototyping with Gemini models, including code generation

#### **v0.dev** - [v0.dev](https://v0.dev)

Vercel's AI tool for generating React components and UI designs from text descriptions

#### **Bolt.new** - [bolt.new](https://bolt.new)

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

### **ChatGPT Codex** - [chatgpt.com/codex](https://chatgpt.com/codex)

OpenAI's code-specialized model integrated into ChatGPT for code generation and understanding

### **Cursor** - [cursor.com](https://cursor.com)

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

### **GitHub Copilot** - [github.com/features/copilot](https://github.com/features/copilot)

AI pair programmer integrated into VS Code, JetBrains IDEs, and other editors

### **Windsurf** - [codeium.com/windsurf](https://codeium.com/windsurf)

AI-powered IDE with Cascade flows for multi-file editing and autonomous task completion

### **Supermaven** - [supermaven.com](https://supermaven.com)

Fast AI code completion with 1 million token context window

### **Aider** - [aider.chat](https://aider.chat)

Command-line AI coding assistant that edits code in your local git repository

### **Replit** - [replit.com](https://replit.com)

Collaborative browser-based IDE with AI assistance, instant deployment, and multiplayer coding

---

## Image Generation & Editing

### **Google Nano Banana** - [aistudio.google.com](https://aistudio.google.com)

One of the newest and best image generation and editing models.

### **Midjourney** - [midjourney.com](https://midjourney.com)

Discord-based platform creating high-quality artistic images from text prompts

### **DALL-E** - [openai.com/dall-e](https://openai.com/dall-e)

OpenAI's image generator for detailed images from natural language

### **Decohere** - [decohere.ai](https://decohere.ai)

Generates characters, images, and videos from text and image prompts

### **Stability AI** - [stability.ai](https://stability.ai)

Creator of Stable Diffusion for multimodal generative AI

### **Catbird** - [catbird.ai](https://catbird.ai) (*Defunct*)

Multi-model generator creating images from 15+ AI models simultaneously

### **Leonardo.ai** - [leonardo.ai](https://leonardo.ai)

Fine-tuned models for game assets, concept art, and production visuals

### **Fooocus** - [GitHub](#)

Open-source Stable Diffusion interface with advanced inpainting

### **AUTOMATIC1111** - [GitHub](#)

Popular web interface for Stable Diffusion with extensive features

### **Adobe Express Background Remover** - [adobe.com](https://adobe.com)

Free AI tool for instant background removal with transparent PNG output

---

## Video Generation & Editing

### **Runway** - [runwayml.com](https://runwayml.com)

Advanced AI video platform with Gen-4 models for text/image-to-video

### **HeyGen** - [heygen.com](https://heygen.com)

AI avatar platform for professional spokesperson videos

**Pika** - [pika.art](https://pika.art)

AI video generator creating cinematic clips up to 10 seconds

**Adobe Firefly Video** - [firefly.adobe.com](https://firefly.adobe.com)

Text/image-to-video with 1080p output and commercially safe generation

**Hailuo AI** - [hailuoai.video](https://hailuoai.video)

Chinese AI video generator creating 5-6 second HD videos

**Google Veo 3** - [deepmind.google](https://deepmind.google)

Advanced video generation with 8-second clips and synchronized native audio

**Google Flow** - [labs.google/fx/tools/flow](https://labs.google/fx/tools/flow)

Google's experimental video generation tool from Google Labs FX

**Hedra** - [hedra.com](https://hedra.com)

Specializes in talking avatar generation with realistic lip-sync

**Synthesia** - [synthesia.io](https://synthesia.io)

AI presenters for training videos and marketing content

**Switchlight** - [switchlight.beeble.ai](https://switchlight.beeble.ai)

AI-powered video relighting tool with PBR maps

**Adobe Express Video Background Remover**

- [adobe.com](https://adobe.com)

Remove video backgrounds with transparent MP4 output

**VideoBGRemover** - [videobgremover.com](https://videobgremover.com)

Professional background removal for videos up to 3 hours

**Innova AI Video Background Removal** -

[Hugging Face](#)

Free background removal and replacement tool

**Higgsfield** - [higgsfield.ai](https://higgsfield.ai)

Video and image generation with a focus on impossible effects and transitions

---

## Audio & Music

**Hume** - [hume.com](https://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](https://descript.com)

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

**ElevenLabs** - [elevenlabs.io](https://elevenlabs.io)

Advanced AI voice cloning and text-to-speech in 30+ languages

**Suno** - [suno.ai](https://suno.ai)

AI music generation with vocals and instruments from text

**Udio** - [udio.com](https://udio.com)

Professional-quality music tracks from text descriptions

**Adobe Podcast Enhance** -

[podcast.adobe.com](https://podcast.adobe.com)

AI audio enhancement removing background noise

---

## AI Chat & Search

**Perplexity** - [perplexity.ai](https://perplexity.ai)

AI-powered search with sourced answers and citations

**Maple.ai** - [trymaple.ai](https://trymaple.ai)

Privacy-focused with end-to-end encryption and confidential computing

**DeepSeek** - [deepseek.com](https://deepseek.com)

Chinese open-source AI with powerful reasoning (R1 model)

**Venice.ai** - [venice.ai](https://venice.ai)

Privacy-focused uncensored AI with local data storage

**Z.ai Chat** - [chat.z.ai](https://chat.z.ai)

Free AI chatbot with presentation generation and multimodal capabilities

## AI Infrastructure & Development

**Replicate** - [replicate.com](https://replicate.com)

Cloud platform for running open-source AI models via API

**Groq** - [groq.com](https://groq.com)

Ultra-fast AI inference with custom LPU chips

**Hugging Face** - [huggingface.co](https://huggingface.co)

The GitHub of AI - hub for models, datasets, and applications

**Chutes.ai** - [chutes.ai](https://chutes.ai)

Decentralized serverless AI compute platform

## 3D & Spatial

**Depth Anything V2** - [GitHub](https://github.com)

Monocular depth estimation for 3D reconstruction

**Luma Labs** - [lumalabs.ai](https://lumalabs.ai)

Interactive 3D scenes using NeRF and Gaussian Splatting

**Hyper3D (Rodin AI)** - [hyper3d.ai](https://hyper3d.ai)

Production-ready 3D models from text or images

**Brush** - [GitHub](https://github.com)

Open-source 3D reconstruction with Gaussian Splatting

**Tripo3D** - [tripo3d.ai](https://tripo3d.ai)

Fast 3D asset generation in ~10 seconds

**COLMAP** - [colmap.github.io](https://colmap.github.io)

Open-source photogrammetry pipeline for 3D reconstruction

**TRELLIS** - [Hugging Face](https://Hugging Face)

Microsoft's 3D generation with Radiance Fields and meshes

**RealityScan** - [realityscan.com](https://realityscan.com)

Professional photogrammetry by Epic Games

**HitPaw 3D (Hitem3D)** - [hitem3d.ai](https://hitem3d.ai)

Ultra-high resolution 3D models from single images

**Postshot** - [jawset.com](https://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](https://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](https://chatgpt.com/atlas)

ChatGPT in a browser with memory and automation abilities

#### **Perplexity Comet** - [perplexity.ai/comet](https://perplexity.ai/comet)

AI-first browser with autonomous task execution and integrated search

#### **Opera Neon** - [operaneon.com](https://operaneon.com)

Premium agentic browser with Chat, Do, and Make functions

#### **Fellou** - [fellow.ai](https://fellow.ai)

Self-driving browser with deep search and agentic memory

#### **Sigma AI Browser** - [sigmabrowser.com](https://sigmabrowser.com)

Privacy-first with end-to-end encryption and Deep Research mode

#### **Dia Browser** - [diabrowser.com](https://diabrowser.com)

Tab-aware assistant with Skills system and natural language commands

---

## Specialized Tools

#### **ComfyUI** - [comfy.org](https://comfy.org)

Node-based workflow platform for multiple AI model types

#### **Cascdr** - [cascdr.xyz](https://cascdr.xyz)

Bitcoin Lightning-powered AI workflow platform

#### **Manus.im** - [manus.im](https://manus.im)

Browser-based AI agent platform for autonomous tasks

---

#### **Genspark** - [genspark.ai](https://genspark.ai)

Many AI tools in one chatbot

#### **Browserbase** - [browserbase.com](https://browserbase.com)

Developer infrastructure for headless browsers and AI agents

---

## 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](mailto:noaheisenbruch@gmail.com)