

Your AI Guide For This Session



Top AI Innovator

Named to Dallas Innovates'2025 AI Maverick



Author, *Generative AI for Everyone: A Practical Guidebook*

Created the CLEAR AI framework to simplify complex AI workflows.



Director of Innovation – Tonic3

Strategy, Product Development, Design, And Beyond



Founder, Agent Insights

Building the future of AI observability for multi-agent systems and behavioral AI evolution.



Expert in AI & UX Strategy

25+ Years In UX and Development. Pioneering agentic systems that keep humans in the design loop.



Speaker & Educator

Co-founder of SMU UX Program, Speaking, Teaching + Workshopping



Preston McCauley

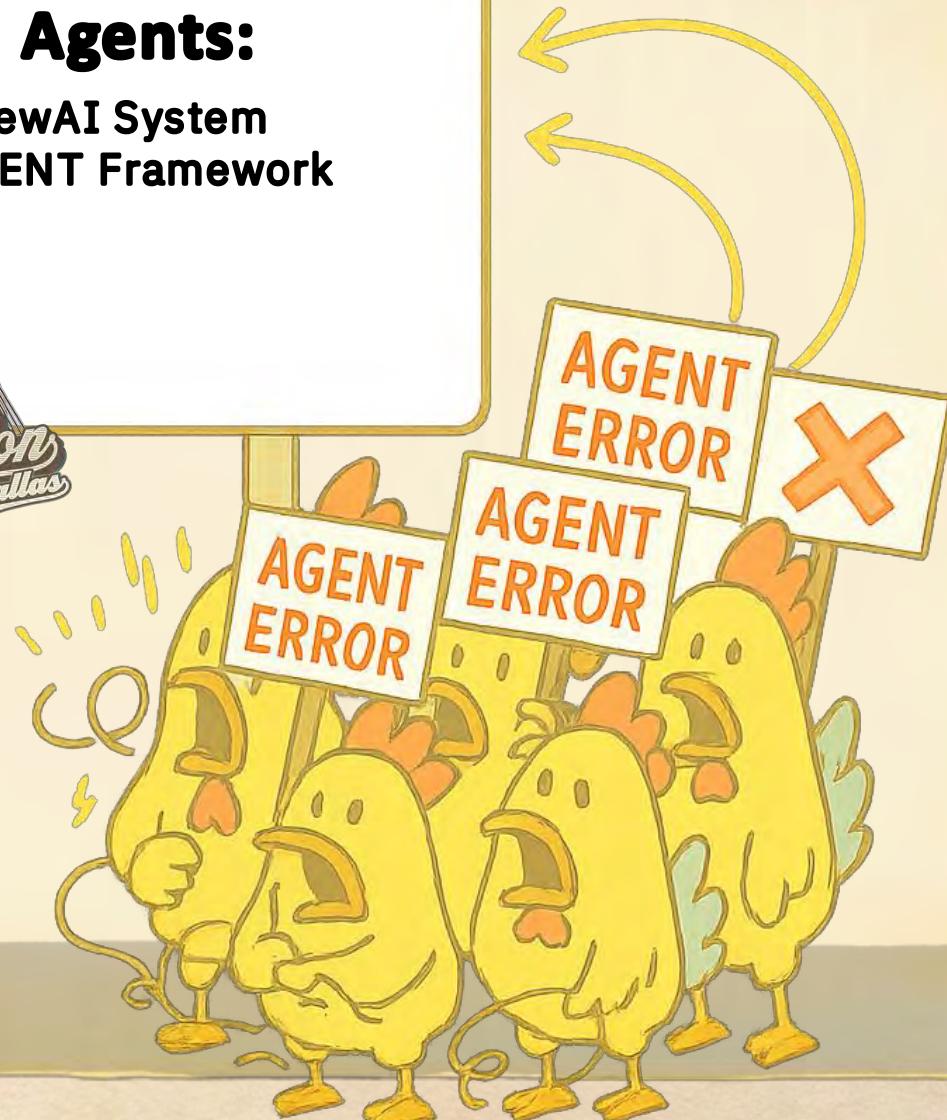
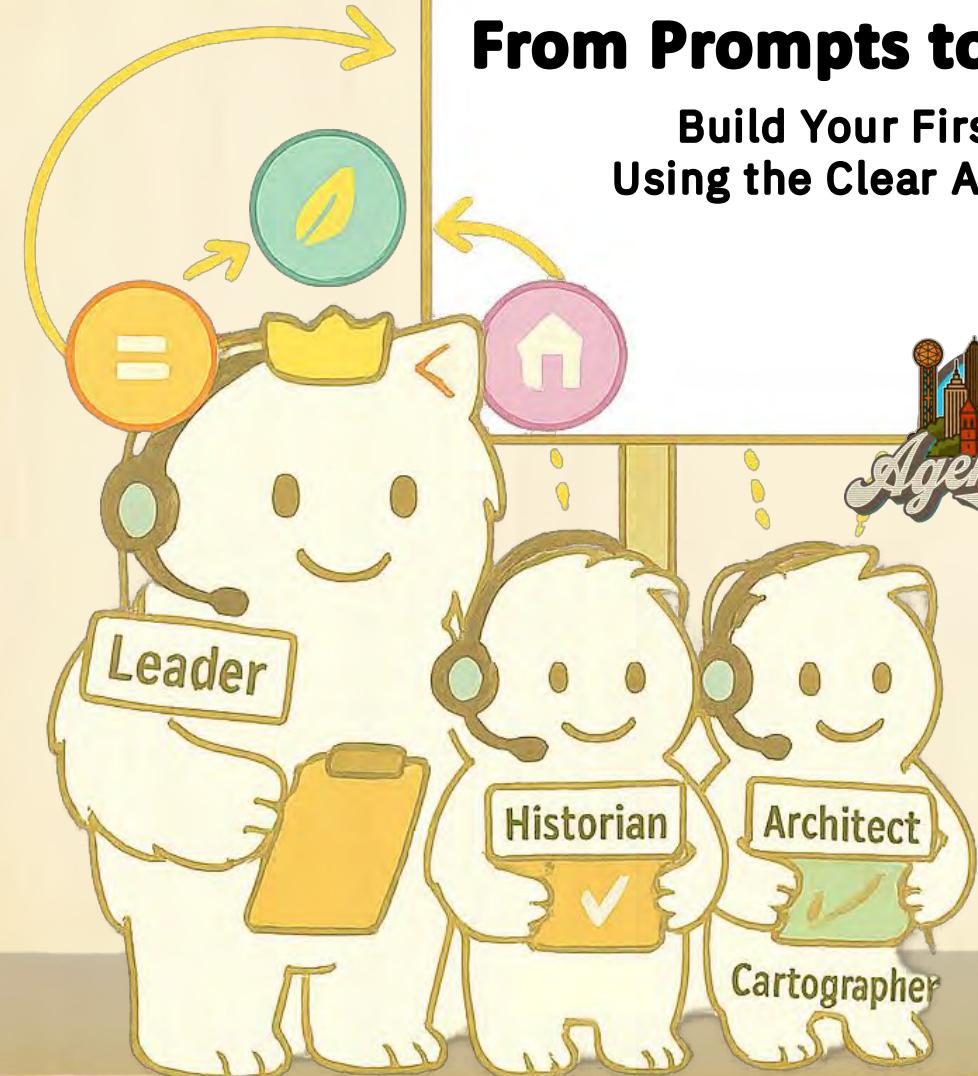
Director of Innovation AI - Tonic 3
AI / UX / AR / VR And Beyond





From Prompts to AI Agents:

Build Your First CrewAI System Using the Clear AI AGENT Framework



The Rise Of Agentic AI

From Tools to Teammates:

AI that “Thinks” before action.

Strength in Numbers:

AIs that work together to complete tasks

Trusting the Unexpected:

Importance of knowing what your agents are doing



The Importance Of Planning Agentic Flows

Agents Are Amazing... When They Work

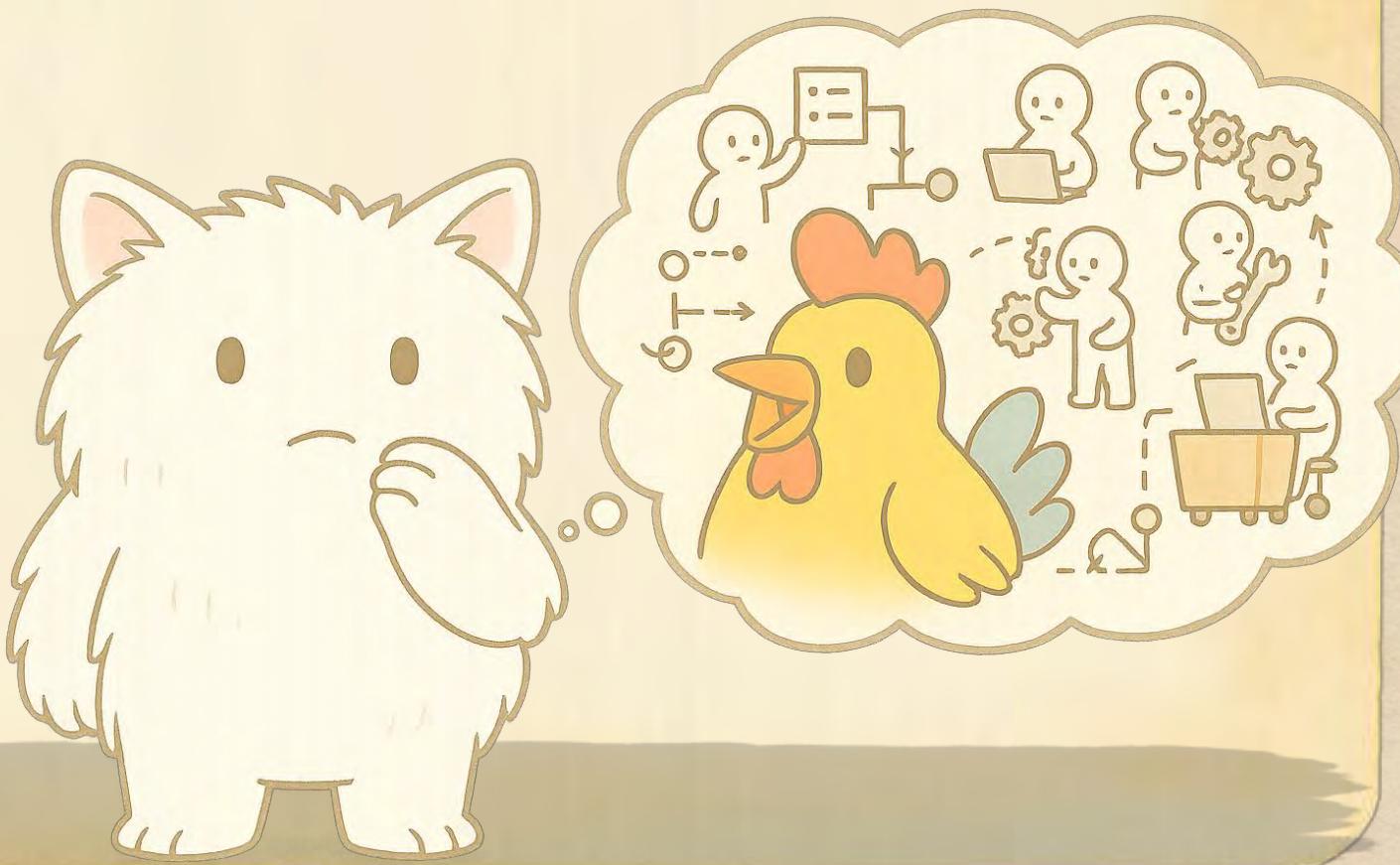
AI that “Thinks” before action.

Planning Your Structures Is Crucial

AIs that work together to complete tasks

Roleplaying Is Instrumental

Importance of knowing what your agents are doing



Planning Agents with Clear AI A.G.E.N.T.S

Assign Roles:

What are the leading roles?

Generate Objectives:

What purpose or meaning is each agent responsible for?

Establish Guard Rails:

Not just what to be done, what not to do!, where not to go.

Note Success Metrics:

What does success look like for each agent in the chain?

Transition Hand Offs:

How does an agent know that its job is done? Please don't leave it to chance!



So What Does CrewAI Bring To The Table?

Lightweight:

Easy to set up, with some pitfalls

Defines Roles, Tasks & Tools:

Allows you to map out a plan for your agents.

Supports Different Modes:

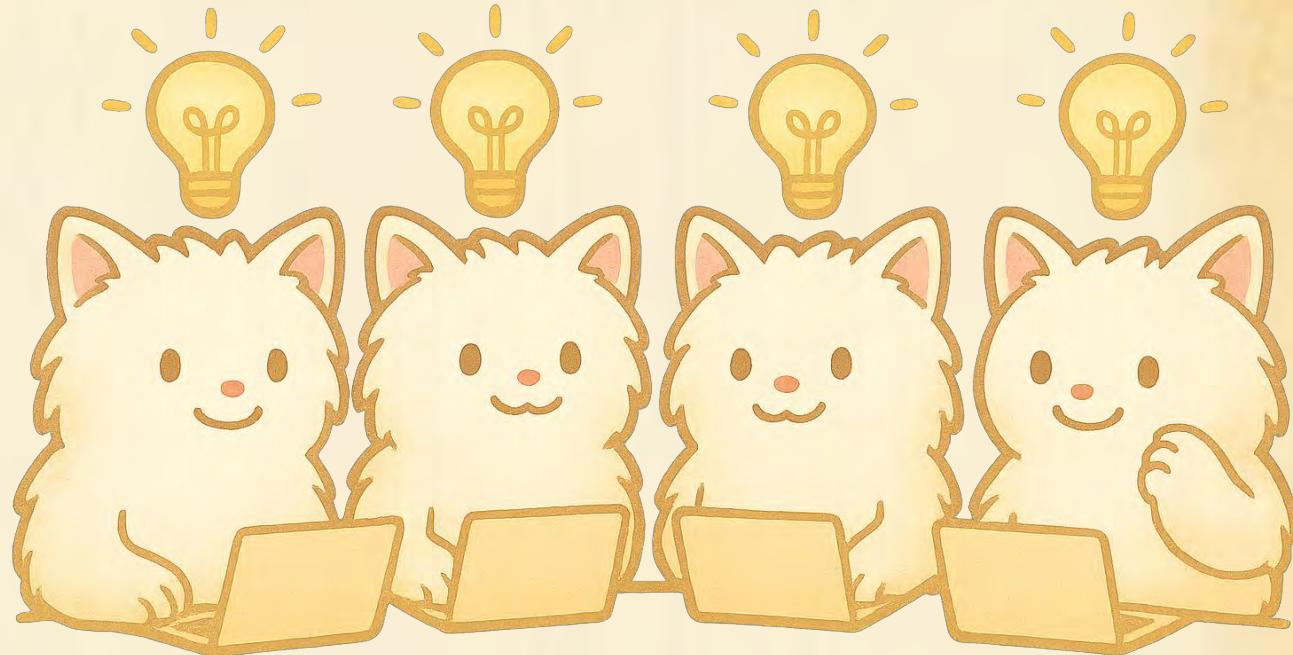
Sequential and hierarchical tasks

Can Run With Nearly Any LLM:

Agentic runs don't need to cost 1000s of dollars!

Enables Quick Iteration:

Python is very adaptable, add your logging, and more.



The Plan For Our History Crew

Purpose: Build a complex agentic structure that can create images that represent a period of time, but with a fictional twist.

SearcherAgent

Retrieve bulk data like maps.

VisionAgent:

Describes visual characteristics and artistic elements based on historical knowledge.

HistorianAgent:

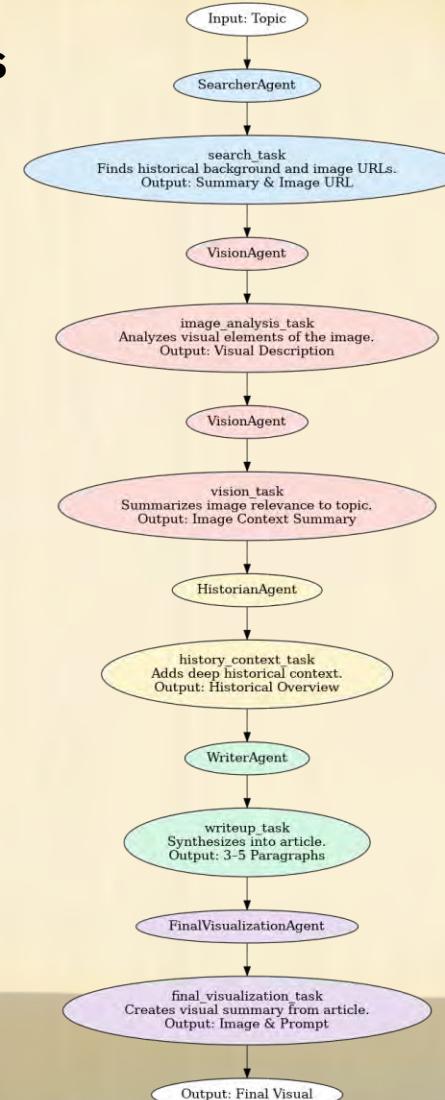
Sequential and hierarchical tasks

WriterAgent:

Provides comprehensive historical analysis and context.

FinalVisualizationAgent:

Creates visualizations that synthesize research findings..



Let's Explore The Code A Bit

Let's Take A Look At Some Code!



What's A Typical CrewAI Construction?

crew.py:

Manages the crew of agents, including their creation, task assignment, and interactions.

main.py:

Serves as the entry point of the application, orchestrating the overall process and workflow.

agents.yaml:

Defines the agents and their capabilities, including roles, tools, and specific configurations.

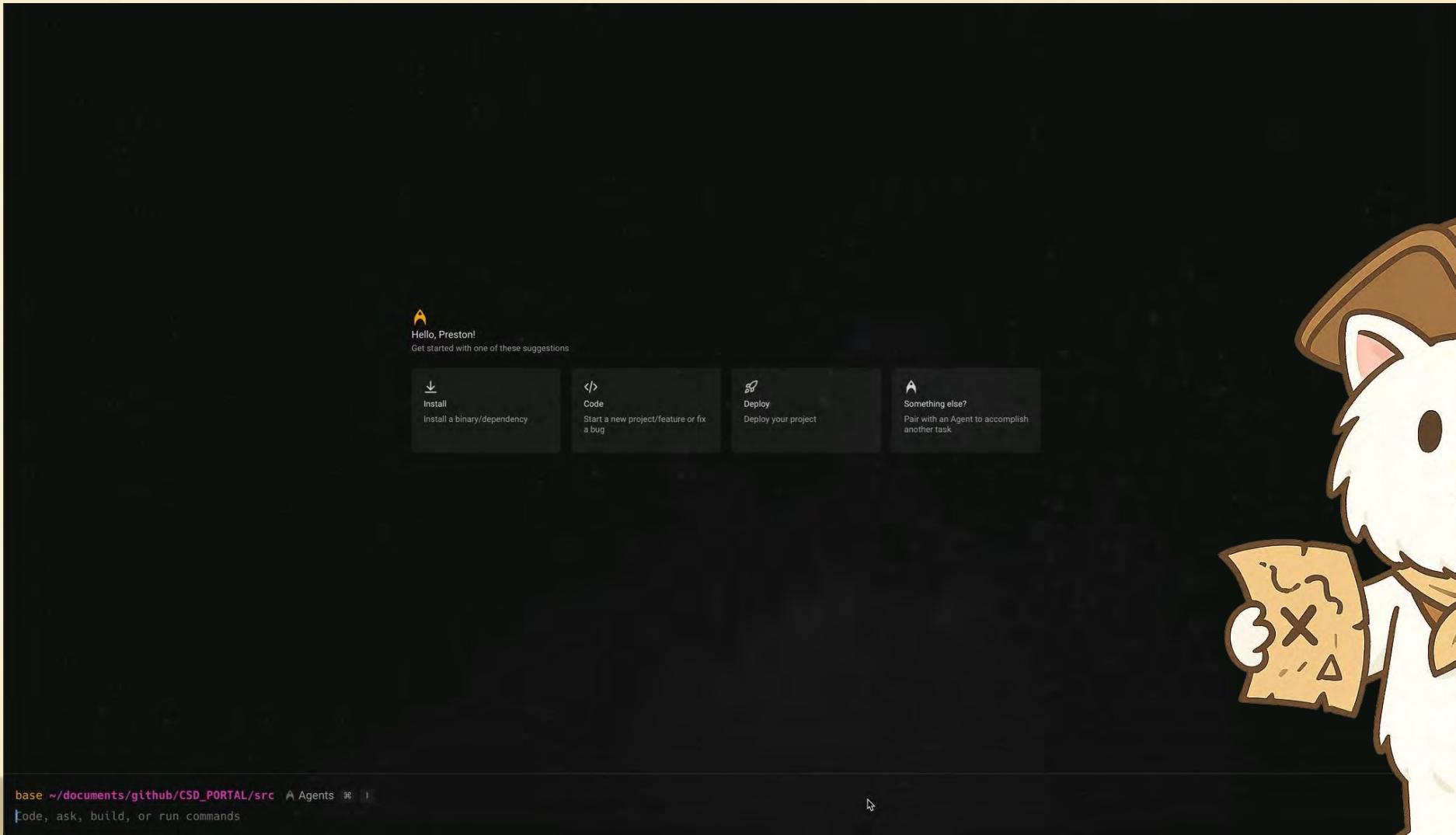
tasks.yaml: Outlines the functions that the agents will perform, including task names, descriptions, and parameters.

tools: Manages the crew of agents, including their creation, task assignment, and interactions.

```
.|__ crew_historian
|   __init__.py
|   __pycache__
|     __init__.cpython-311.pyc
|     crew.cpython-311.pyc
|   config
|     agents.yaml
|     tasks.yaml
|   crew.py
|   main.py
|   tools
|     __init__.py
|     __pycache__
|       __init__.cpython-311.pyc
|       mistral_image_creation_tool.cpython-311.pyc
|       mistral_vision_tool.cpython-311.pyc
|       scraping_dog_search_tool.cpython-311.pyc
|     check_image.py
|     custom_tool.py
|     debug_image_download.py
|     debug_mistral.py
|     mistral_image_creation_tool.py
|     mistral_vision_tool.py
|     scraping_dog_search_tool.py
6 directories, 19 files
```



Let's Setup Crew



Let's Take A Look At Our Historian Crew



```
crewai21 ~/documents/github/crew_historian_demo git:(main) ✘ ±8  Agents  %  i  
PYTHONPATH=src python src/crew_historian/main.py test "The Rise Of The Baroque Movement In Art" ⏎→ ⏎
```

Why Do I Like Crew AI?

Best Self-Directed Agentic AI:

It's best to set and forget if you plan.

Train Mode:

Creates a run sequence to help fine-tune and provide Human feedback before the whole run.

Readable roles & tasks:

Clear role and task definitions

Quick prototypes:

Fast prototyping and iteration

Swap tools / LLMs fast:

Supports Local Ollama and LiteLLM under the surface.

Run Locally:

Run In An isolated environment.

Spin-up Your Own Tools:

Need a tool to create one, connect to an MCP, Vision Models, more...



Tell Me Something Bad About CrewAI

Architecture & Execution

Lacks async orchestration: No parallel task support

Sparse observability & traces:

No native run insights

No persistent memory:

No state between runs

Manual YAML management:

Error-prone static config

Rigid flow control:

This may have been solved with crew FLOW now.

Fragile UVTool binding:

Breaks across agents



From Demo's To Deployment

Start small, scale smart:

Begin with small projects and scale intelligently.

Cost management is crucial:

Consider implementing guardrails to help agents mitigate over-tooling usage.

Add memory & async:

Incorporate memory and asynchronous capabilities.

CrewAI Enterprise:

Works well, but you need to use both UV and GitHub to benefit.

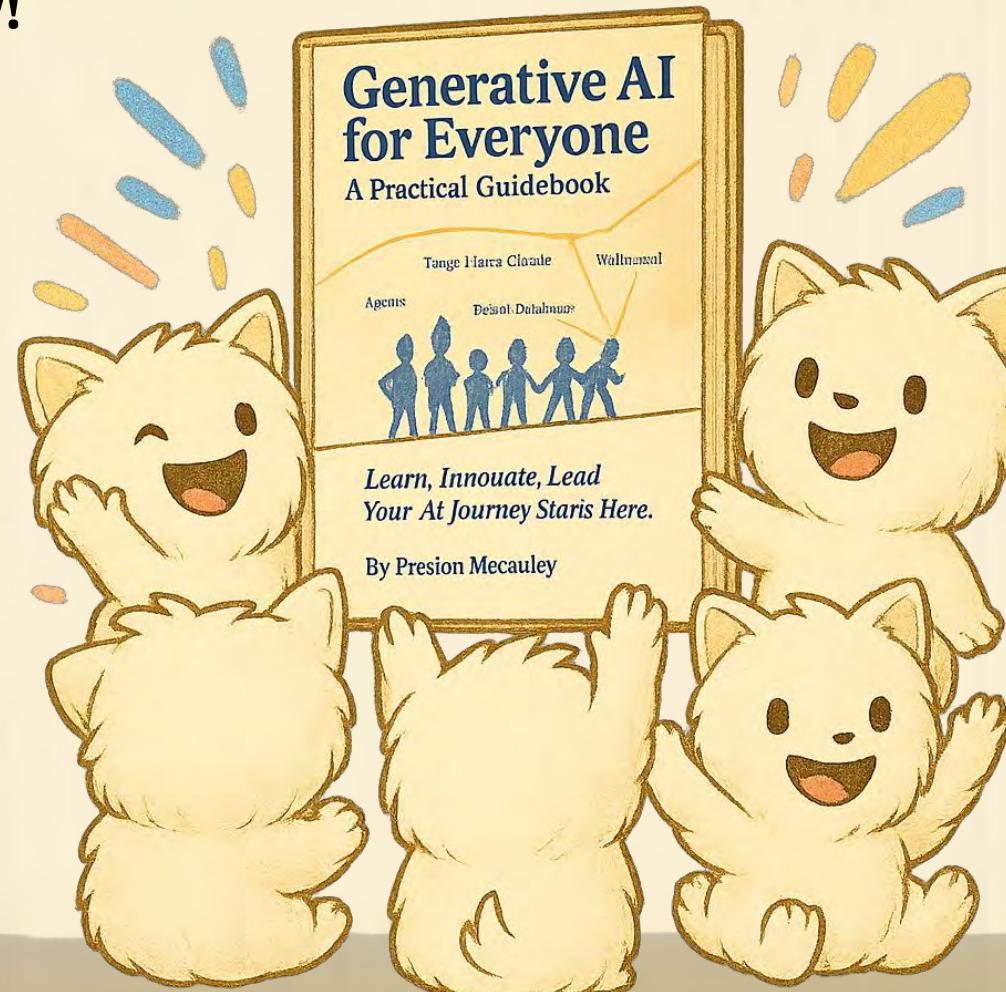
CrewAI works Best With OpenAI In Production:

Certain plugins require OpenAI currently in tools, not decoupled.



Thank You – Happy Creating

Get My Book Available Now!



Preston McCauley

Director of Innovation AI - Tonic 3
AI / UX / AR / VR And Beyond

