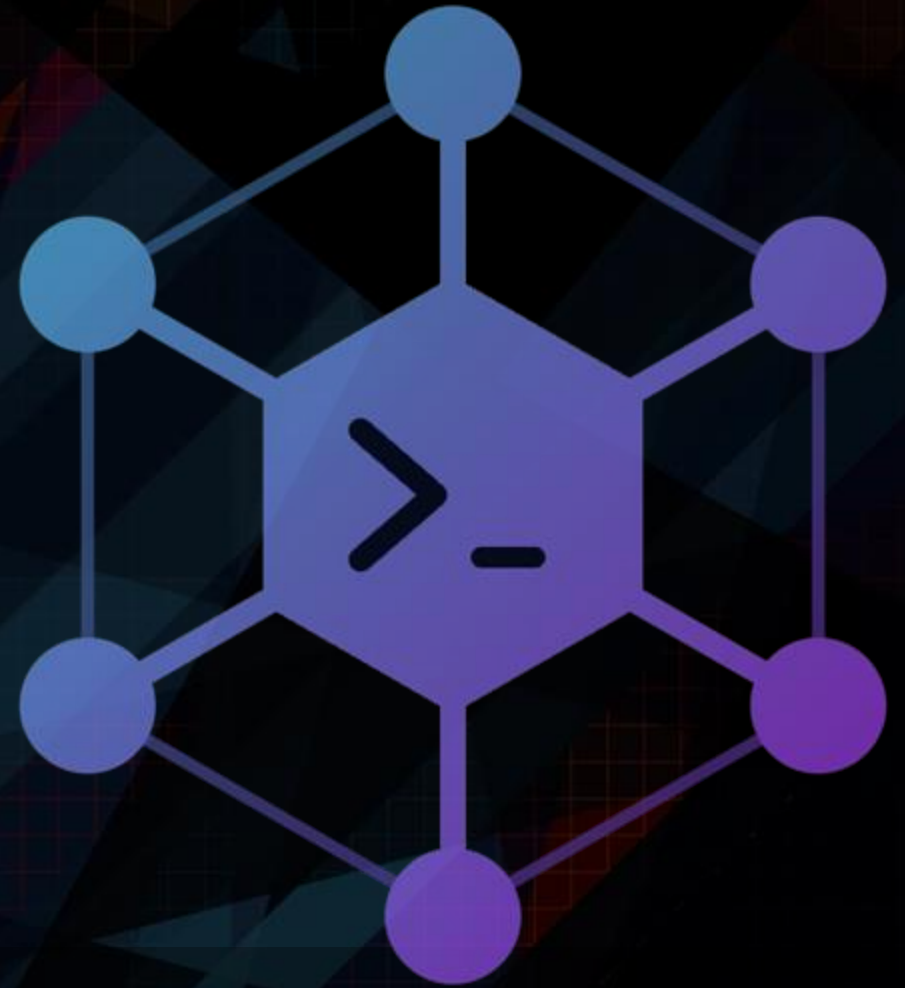


AI Task Agent Using More Flexible Generic Tools via Model Context Protocol (MCP)

Milestone 2

*Jadon Duff
Jacob Ewasko
Ioannis Limniatis
Jacob Lebkuecher*



Milestone 2 Progress Report

Task	Completion %	Jadon	Jacob E	Jacob L	Ioannis	To Do
1. Implement, test, and demo Prototype for Task Execution using Generic MCP Tools, without advanced security features implemented (feature 1).	100%	45%	25%	15%	15%	None
2. Implement, test, and demo Prototype AI Agent, with full connection to Prototype Task Execution using Generic MCP Tools (feature 1).	100%	30%	30%	20%	20%	None
3. Implement, test, and demo POC RAG memory, with basic documentation tested and linked to Prototype AI Agent (feature 2).	100%	40%	20%	20%	20%	None
4. Implement, test, and demo Prototype for Versatile Deployment with Widespread Hardware Compatibility, with the system functional to current development on a new install on a new system (feature 3).	100%	25%	25%	25%	25%	None



Task 1

- **31 generic MCP tools were implemented** such as tools for shell, Python, browser, screen interaction, file CRUD, web search, Brain memory, and environment info.
- **Brain feature provides persistent cross-session memory** — agent remembers and learns across sessions
- **Browser suite MCP tools built on Playwright** — semantic ARIA-based navigation for reliable web interaction
- **37 of 131 requirements verified** via structured test cases
- **Obstacle:** Smaller LLMs failed to invoke tools → resolved with higher-parameter models

AI Task Agent helped with this slide



New chat

Search

Models

MCP

Settings

Recent conversations

Video Subtitle Script Creation

Create Basic HTML in VS Code

Running send_email tool without re...

Send Email Tool Password Error

Requesting email composition details

Create HTML File Using VS Code

Create HTML in VS Code

Creating HTML File Using VS Code

Creating Basic HTML via VS Code

Secret Message in Transfer Folder

GitHub Repository Self-Cloning and...

AI Task Agent Clone and Execution

Sum Numbers Left of Slash Python

Milestone 2 Presentation Slide 1 Bul...

Comprehensive Overview of Availab...

Writing Hello World Python File

List of available AI tools and functions

List of Available AI Assistant Tools

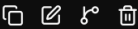
AI Assistant Tool Capabilities Overv...

Hello, I would like you to create a basic HTML file. Do this by downloading VS Code for ARM64 in VS Code, and typing the HTML into a VS Code tab. Then, show me the final product in our shared browser after you finish editing in VS Code.

21,929 tokens

1min 46s

206.75 tokens/s



Reasoning

remember

87 tokens

5.4s

16.21 t/s

Reasoning

recall

recall

72 tokens

3.6s

19.79 t/s

Reasoning

Hey Jadon! Let me get VS Code set up and create that HTML file for you. First, let me check the system architecture.

execute_command

204 tokens

12s

16.55 t/s

Reasoning

Confirmed — we're on ARM64. Downloading VS C now.

type a message...

+

Medium

Qwen3.8 27B

Task 2

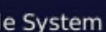
- **End-to-end agentic loop operational** — LLM reasons, selects tools, executes, and returns results.
- **Virtual Computer Developed & Implemented** — LLM has root access to an Ubuntu Docker container
- **System prompt engineered** for tool selection, task decomposition, and Brain usage.
- **Per-member LLM optimization** — models tuned to each member's hardware (7B–27B range).
- **Obstacle:** Trial-and-error across heterogeneous hardware; some models fast but inaccurate, others crashed.
- **Adopted llama.cpp built-in frontend + proxy** to eliminate custom frontend work.

AI Task Agent helped with this slide





Trash



File System

[Home](#)

scratch transfer

2 folders | Free space: 817.0 GiB

AI Task Agent



Task 3

- **EmbeddingGemma2 model validated** — full pipeline: ingest → embed → store → query → retrieve
- **Tested with live and synthetic documents**; relevant sections correctly returned (e.g., "supersonic flight")
- **Obstacle:** llama.cpp version lacked embedding model support → fixed with latest build
- **Foundation set for Milestone 3** — will expand to PDFs, DOCX, video, and full agentic access

AI Task Agent helped with this slide



Task 4

- **One-command start/stop on macOS and Windows** via Makefile + OS-specific scripts
- **Three-port architecture:** 6573 (Web App/LLM), 6574 (MCP server), 6575 (Agent's VM)
- **Installation optimized:** ~1 hr / 36 GB → ~15 min / 3.6 GB via package pruning
- **Docker container isolates agent environment** with controlled /transfer, persistent /brain, temp /scratch
- **Access to containerized VM**
- **Linux support confirmed by AI Task Agent, taking ~17 minutes in all.**

"Hello. Your code is located at the following GitHub link:

<https://github.com/jadonduff/ai-task-agent>

AI Task Agent helped with this slide

*You will clone yourself into your own computer and run yourself.
Please implement"*



Demo

- [CSE 4104 AI Task Agent Demo: HTML page creation](https://www.youtube.com/watch?v=pvfcEMv7YJE) (https://www.youtube.com/watch?v=pvfcEMv7YJE)



AI Task Agent Accomplishments

- Created subtitles for demo video by watching 13-minute recording
- Self-replicated itself by cloning its own code and building another version of itself on its computer
- Created basic applications by downloading VS Code from the web and clicking through GUI menu
- Helped us create computer wallpaper by suggesting, downloading, and setting up Inkscape
- Gave “User Feedback” based on the tools we exposed to it and modifications it wanted for better “UX”
- Created a 3D model of an aircraft with no help
- Identified the correct video game from a single screenshot
- Created a presentation about itself
- Developed a quantum simulation to beat $O(n)$ time for searching in an unsorted array (simulation found item in unsorted array in $O(\sqrt{n})$ time, implementing Grover’s Algorithm in Numpy)
- Solved Computer Vision Homography matrix practice questions from just screenshots

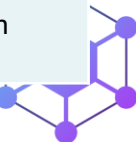
System Requirements Update

		ID	Complete / Count	% Done	Meaning
Functional Requirements		FTE	17 / 39	43.6%	Functional Task Execution using Generic MCP
		FAI	0 / 9	0.0%	Functional Improved AI Knowledge
		FDP	7 / 20	35.0%	Functional Versatile Deployment
Interface Requirements		IUI	6 / 16	37.5%	Interface User Interface
		ISI	0 / 6	0.0%	Interface Software Interface
Performance Requirements		PFR	4 / 13	30.8%	Performance
		OPS	3 / 28	10.7%	Task Based Performance (Operations)
		ATA	37 / 131	28.2%	Total



Milestone 3 Task Matrix

Task	Jadon	Jacob E	Jacob L	Ioannis
1. Implement, test, and demo <u>MVP</u> for MCP tools, meeting <u>100%</u> of the ATA-FTE requirements	Continue leading development, adding MCP tools as the team sees fit.	Optimize browser and Internet tools for the AI and develop the email system.	Fix bugs in the existing tools. Also, implementing performance metrics and clipboard MCP tools.	Optimize token usage, adding token security checks within MCP tools, and developing scheduled task MCP tools.
2. Implement, test, and demo <u>MVP</u> AI Agent and meet <u>50%</u> of <u>all requirements</u> .	Optimize system prompt and all text provided to the LLM, and performing tests for requirements.	Implement authorization to the agentic system and subsystems.	Handle audio ingestion for the Agentic system.	Handle max token and any memory management this team member sees fit.
3. Implement, test, and demo <u>Prototype</u> RAG and meet <u>33%</u> of the FAI requirements.	Enable agentic access to the RAG system, after it is implemented.	Integrate the RAG system with the current architecture, exposing to the LLM (with Jacob L).	Integrate the RAG system with the current architecture, exposing to the LLM (with Jacob E).	Perform tests of the RAG system.
4. Implement, test, and demo <u>MVP</u> for Deployment System and meet <u>50%</u> of the FDP requirements.	Maintain the architecture this team member implemented from the previous Milestone, optimizing Docker container build and adding any relevant additional tools, and developing the Tether system.	Investigate request interception via proxy. Development of a POC or leading a discussion detailing why the current methodology works best will count as equal contribution to development for this Task (with Ioannis).	Optimize the deployment and running of the system, and will create a POC for automated installation. Team member will develop for Windows.	Investigate request interception via proxy. Development of a POC or leading a discussion detailing why the current methodology works best will count as equal contribution to development for this Task (with Jacob E).



Milestone 3 Tasks

- Task 1 – Implement and test and demo MVP for Task Execution using Generic MCP Tools. Ensuring a minimum of 100 Percent. Completing all 39 requirements.
- Task 2 – Implement, test, and demo the MVP AI Agent, with full connection to Prototype Task Execution using Generic MCP Tools. Features should complete at minimum 50% of all requirements in the document, meaning 66 total requirements finished.



Milestone 3 Tasks

- Task 3 – Implement, test, and demo Prototype RAG memory; will meet at a minimum of 33 percent of the ATA – FAI requirements laid out in the documentation
- Task 4 – Implement, test, and demo MVP for Versatile Deployment with widely adaptable Hardware. At minimum 50 percent of the ATA-FDP requirements in the documentation will be fulfilled, meaning 10 of the 20 laid out will be completed.



Questions

*AI Task Agent –
Milestone 2*

