Kasadara

Local LLM, Agentic AI Model Evaluation, Performance Tuning, and Web-Assisted Research Using LM Studio

LM Studio

Kasadara

Table of Content

Kasadara

1. Executive Summary

- Designed and deployed a fully local LLM and agentic AI environment using LM Studio, ensuring cost efficiency, security, and operational control
- Evaluated and optimized open-weight foundation models using industry benchmarks, public leaderboards, and hands-on performance tuning
- Implemented advanced inference optimizations, including GPU offloading and speculative decoding, to improve latency and throughput
- Built agentic and web-assisted research workflows with tool and MCP integrations for scalable automation
- Demonstrated vision-language intelligence through accurate extraction of structured data from real-world financial documents

2. Key Purposes

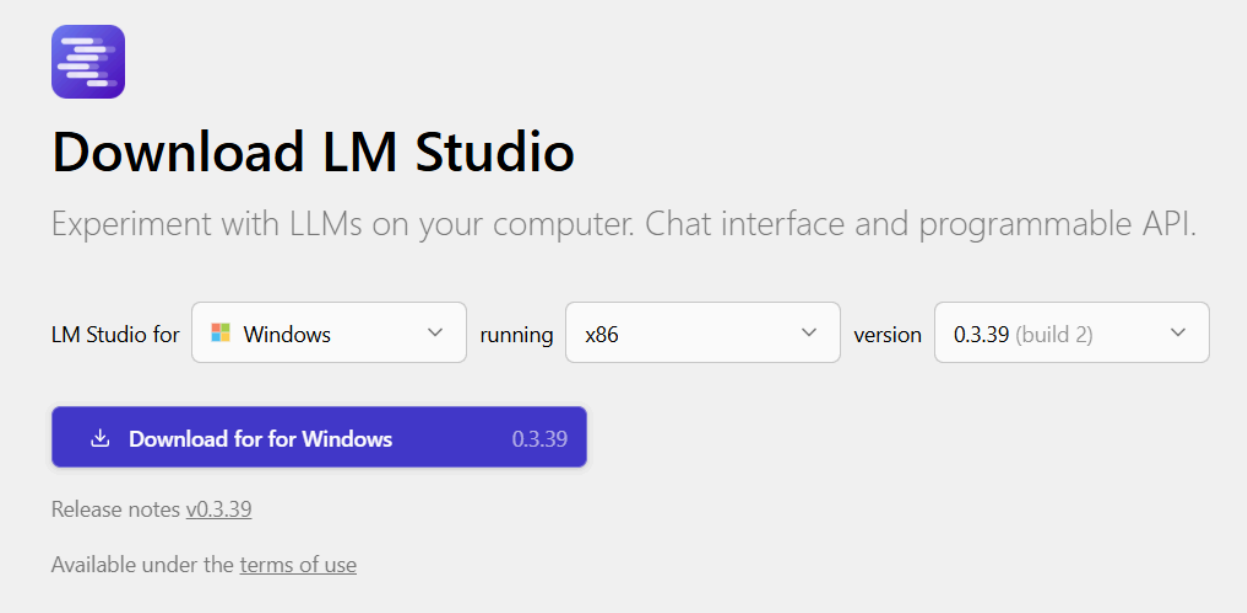
- Design and validate a fully local LLM and agentic AI environment using LM Studio
- Evaluate open-weight foundation models through benchmark-driven and hands-on analysis
- Optimize inference performance using runtime tuning and speculative decoding techniques
- Implement agentic and web-assisted research workflows with tool and MCP integrations
- Demonstrate vision-language capabilities through real-world document data extraction

3. Local LLM Environment Setup (LM Studio)

3.1 System Requirements & Download

Go to the downloads page and click on the appropriate Operating System version that is applicable to your case:

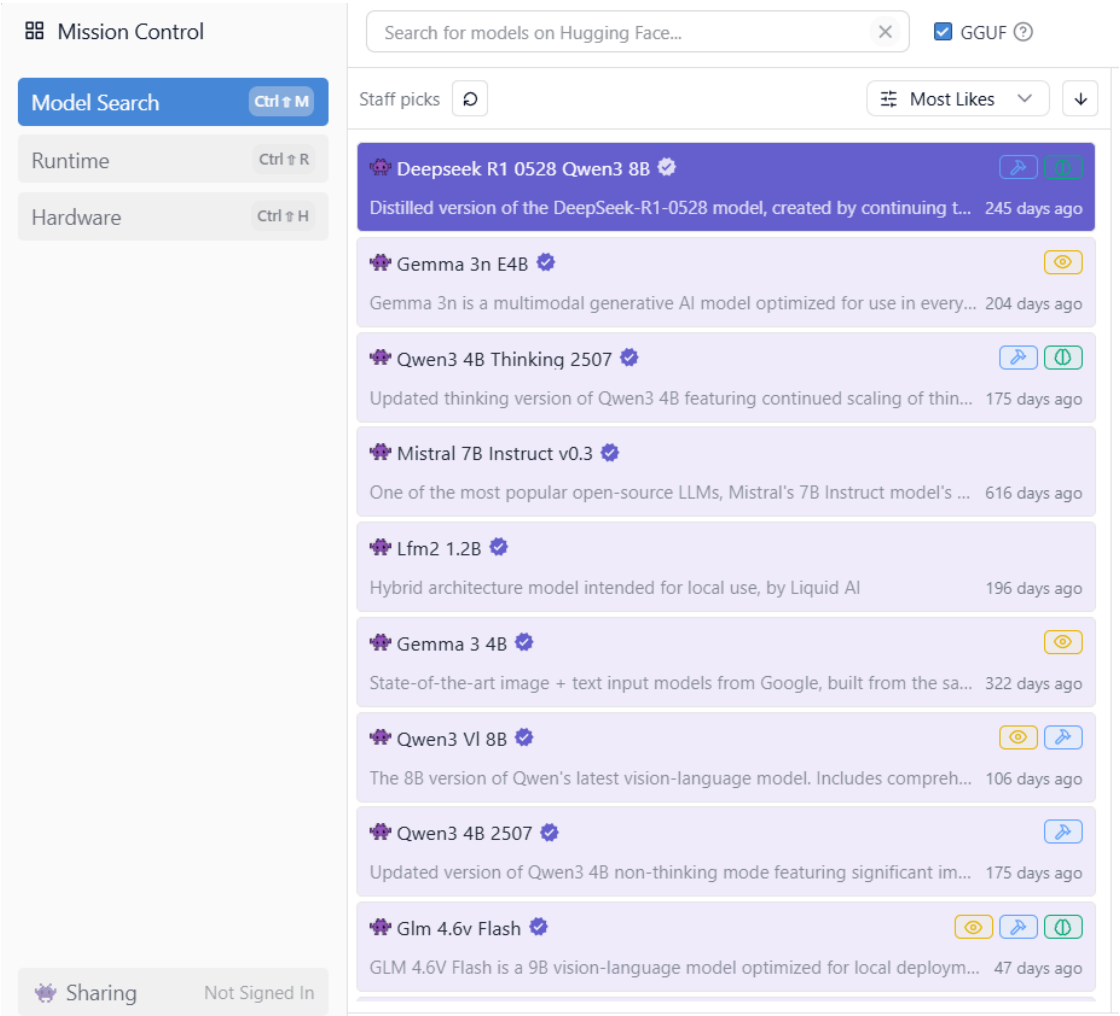
- **Source:** Download the official installer from https://lmstudio.ai/download
- **OS Support**: Available for macOS (Apple Silicon), Windows, and Linux.
- **System Requirements**: LM Studio runs on macOS (Apple Silicon), Windows (x64/ARM), and Linux (x64/ARM64) with 16GB RAM recommended.



With the installation completed, LM Studio is now ready to be launched on the local system.

3.2 Setup & Configuration

After installation, the first step is to download a model. Use the search (Discover) icon on the left panel to browse and select available models.



Select the most appropriate model for your use case; the model selection strategy is discussed in the next section.

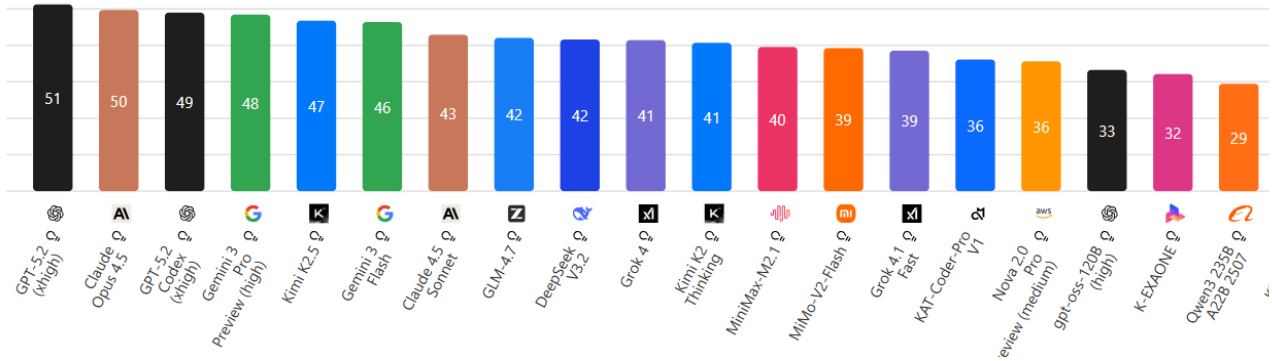
4. Model Benchmarking and Evaluation

4.1 Artificial Analysis

The following benchmarks are sourced from https://artificialanalysis.ai and compare models based on intelligence, agentic capabilities, and more.

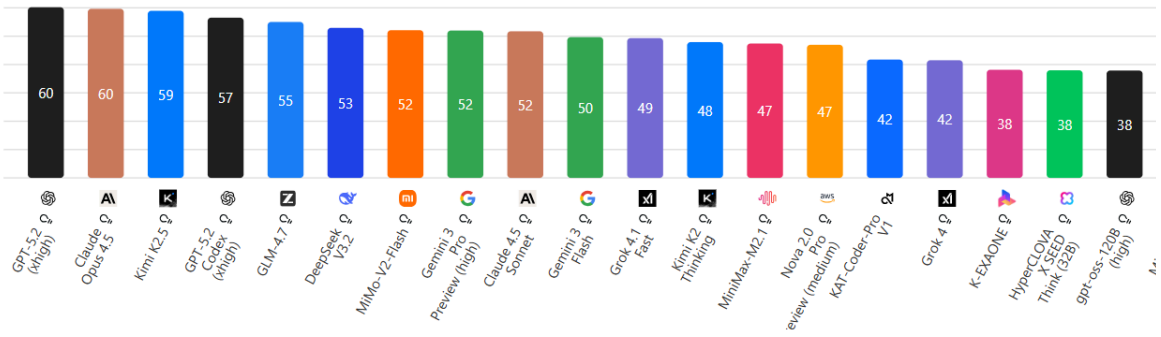
Artificial Analysis Intelligence Index

Artificial Analysis Intelligence Index v4.0 incorporates 10 evaluations: GDPval-AA, τ^2-Bench Telecom, Terminal-Bench Hard, SciCode, AA-LCR, AA-Omniscience, IFBench, Humanity's Last Exam, GPQA Diamond, CritPt



Artificial Analysis Agentic Index

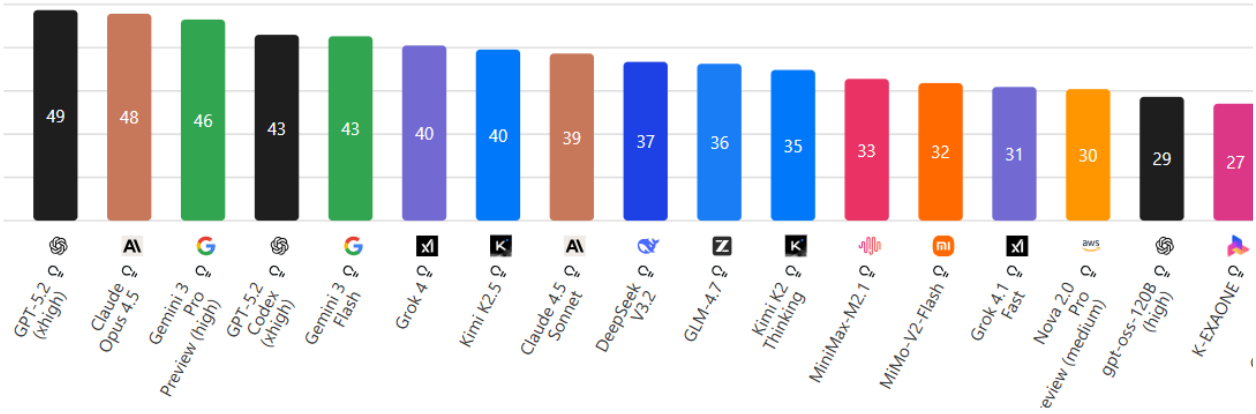
Represents the average of agentic capabilities benchmarks in the Artificial Analysis Intelligence Index (GDPval-AA, τ^2-Bench Telecom)



Kasadara

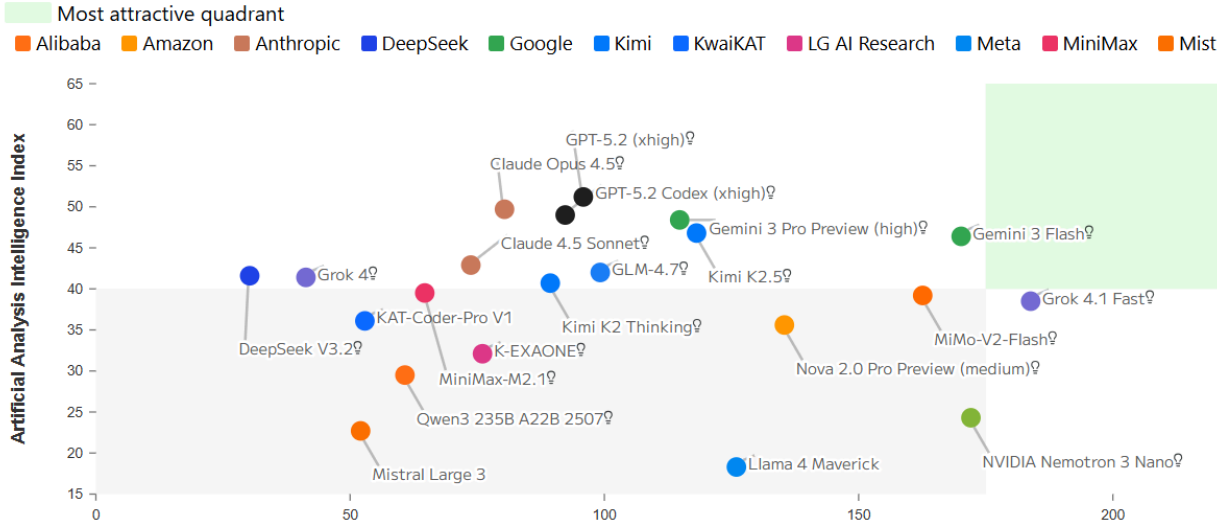
Artificial Analysis Coding Index

Represents the weighted average of coding benchmarks in the Artificial Analysis Intelligence Index (Terminal-Bench Hard, SciCode)



Intelligence vs. Output Speed

Artificial Analysis Intelligence Index; Output Speed: Output Tokens per Second

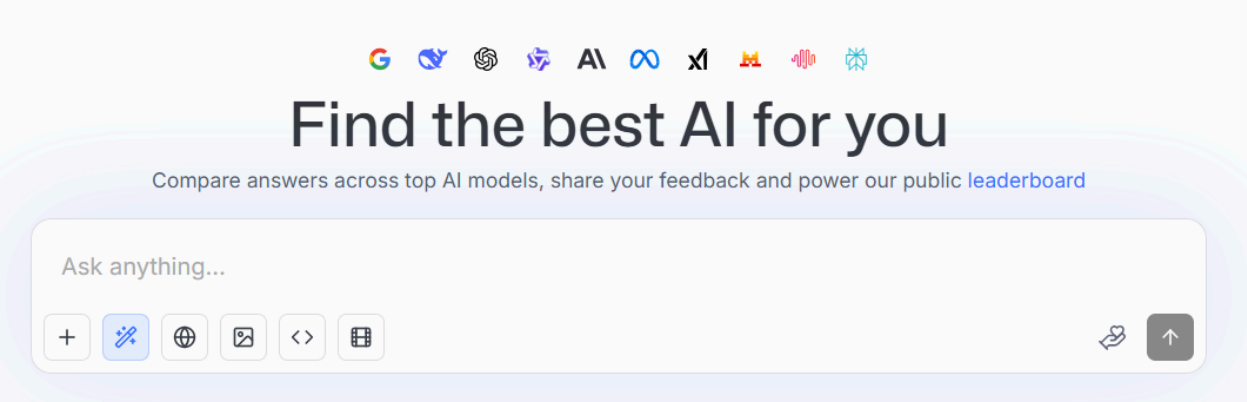


Kasadara

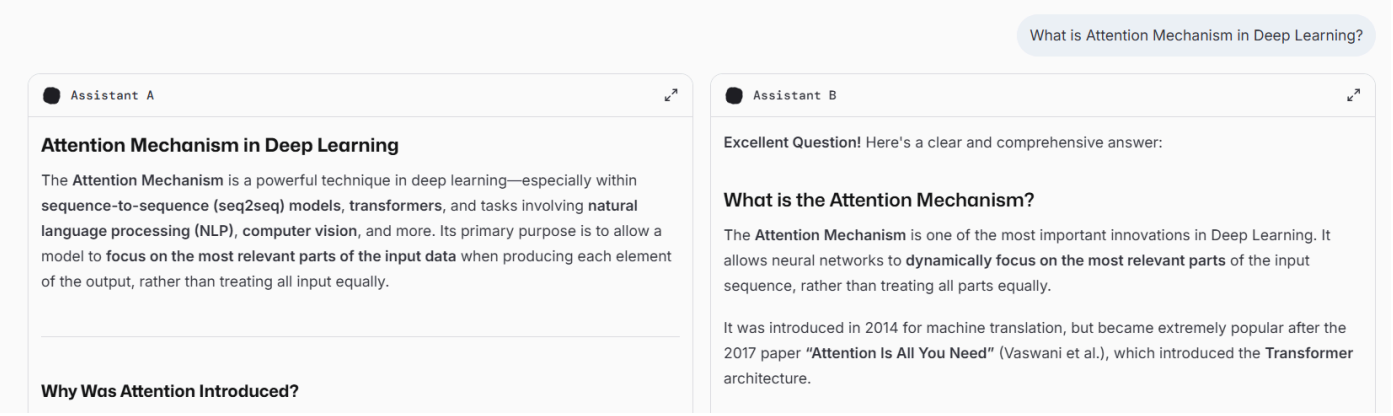
Click on a model to view more details

4.2 LMArena

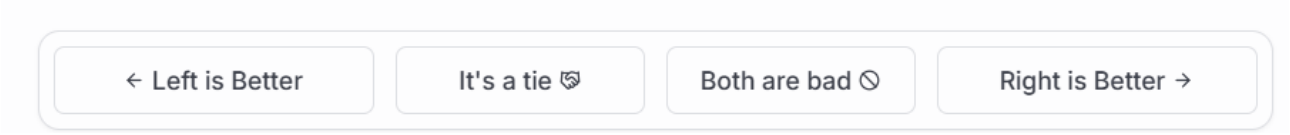
This platform https://lmarena.ai enables side-by-side comparison of answers from leading AI models



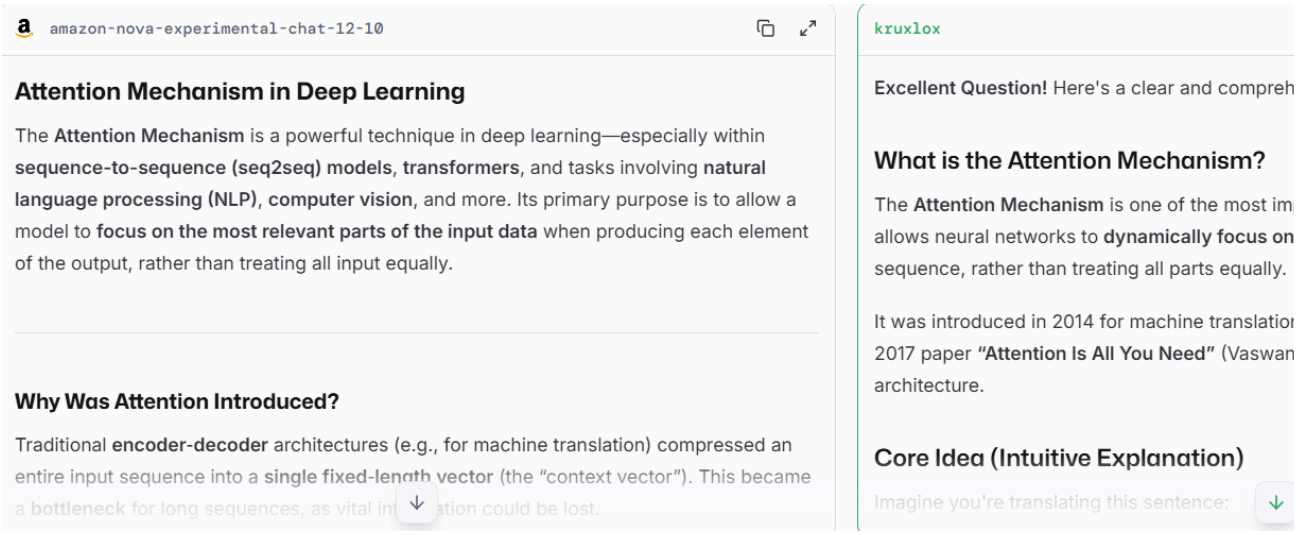
The following section demonstrates a side-by-side comparison of AI-generated responses, along with user-driven evaluation options



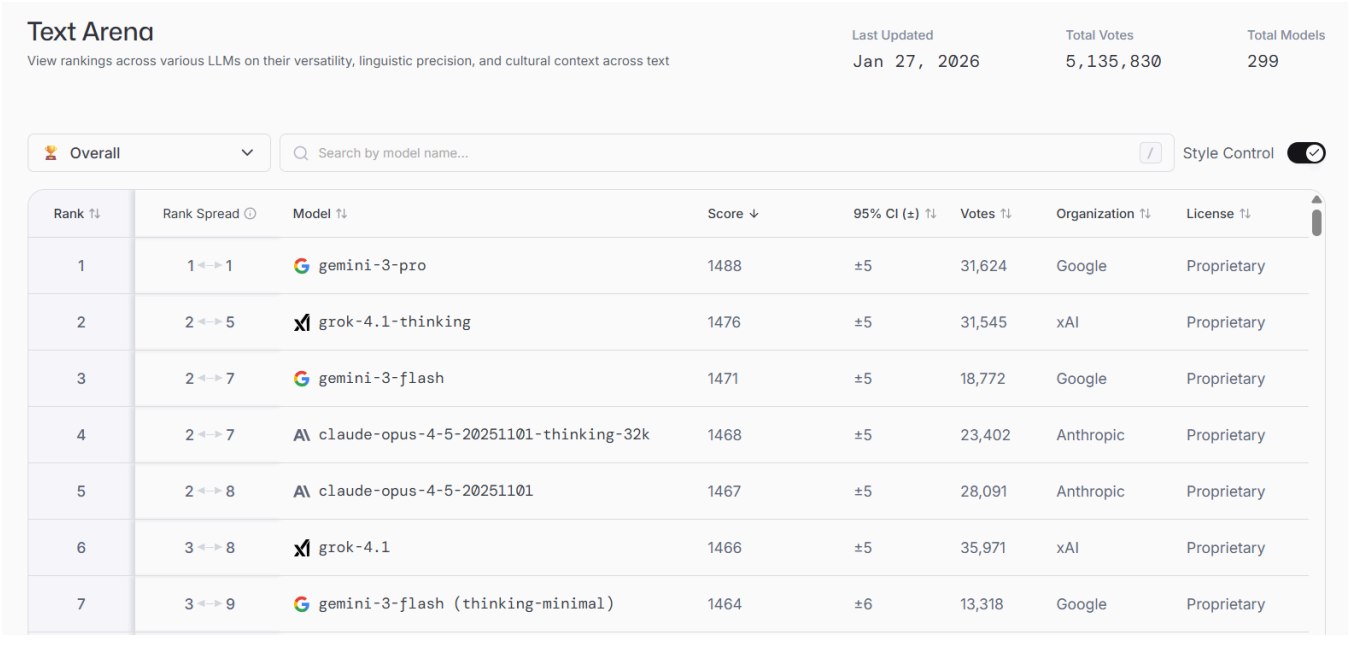
Kasadara



The following image compares responses from two different AI models for the same prompt



LM Arena also provides a public leaderboard ranking LLMs based on community evaluations and performance



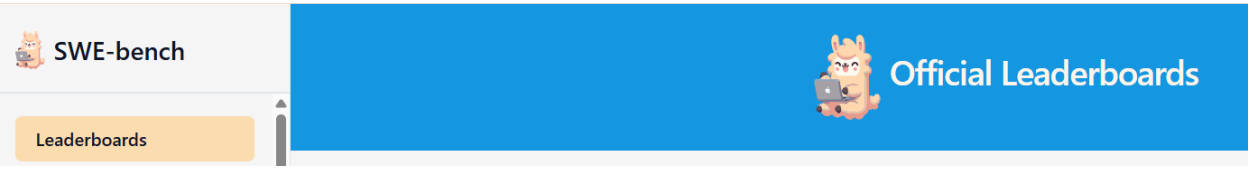
Rank	Rank Spread	Model	Score	95% CI (±)	Votes	Organization	License
1	1 ↔ 1	gemini-3-pro	1488	±5	31,624	Google	Proprietary
2	2 ↔ 5	grok-4.1-thinking	1476	±5	31,545	xAI	Proprietary
3	2 ↔ 7	gemini-3-flash	1471	±5	18,772	Google	Proprietary
4	2 ↔ 7	claude-opus-4-5-20251101-thinking-32k	1468	±5	23,402	Anthropic	Proprietary
5	2 ↔ 8	claude-opus-4-5-20251101	1467	±5	28,091	Anthropic	Proprietary
6	3 ↔ 8	grok-4.1	1466	±5	35,971	xAI	Proprietary
7	3 ↔ 9	gemini-3-flash (thinking-minimal)	1464	±6	13,318	Google	Proprietary

Kasadara

4.3 SWE-Bench

SWE-bench is a benchmark that evaluates large language models on real-world software engineering tasks by testing their ability to understand, modify, and fix code in real GitHub repositories.

Website: https://www.swebench.com



The following table shows SWE-bench leaderboard results comparing models by issue resolution performance.

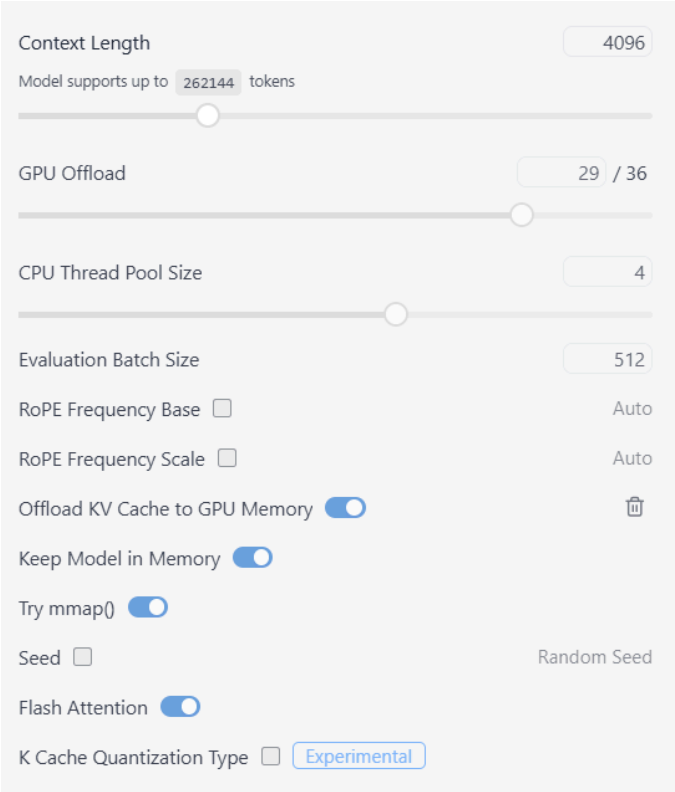
Compare results Filters: All Tags

Model	% Resolved	Avg. $	Trajs	Org	Date	Release
Claude 4.5 Opus medium (20251101)	74.40	$0.72			2025-11-24	1.16.0
Gemini 3 Pro Preview (2025-11-18)	74.20	$0.46			2025-11-18	1.15.0
GPT-5.2 (2025-12-11) (high reasoning)	71.80	$0.52			2025-12-11	1.17.2
Claude 4.5 Sonnet (20250929)	70.60	$0.56			2025-09-29	1.13.3
GPT-5.2 (2025-12-11)	69.00	$0.27			2025-12-11	1.17.2
Claude 4 Opus (20250514)	67.60	$1.13			2025-08-02	1.0.0
GPT-5.1-codex (medium reasoning)	66.00	$0.59			2025-11-24	1.16.0
GPT-5.1 (2025-11-13) (medium	66.00	$0.31			2025-11-20	1.15.0

5. Runtime Configuration and Performance Tuning

The following panel allows users to configure runtime and performance settings for the selected local LLM

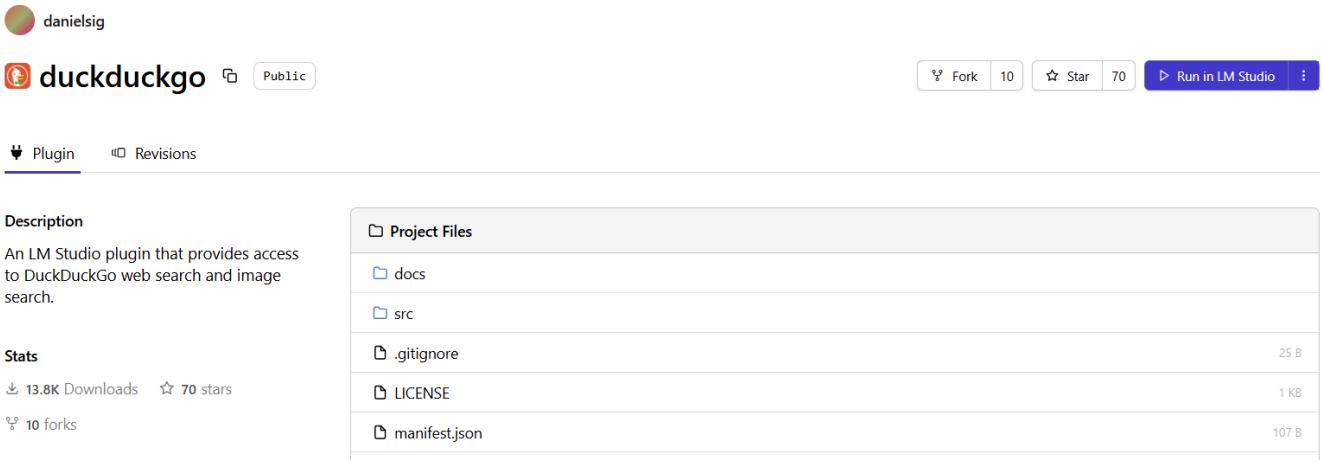
- **Context Length**: Defines how much input the model can process at once.
- **GPU Offload**: Improves performance by moving model layers to the GPU.
- **CPU Thread Pool Size**: Controls CPU parallelism during inference.
- **Evaluation Batch Size**: Impacts inference speed and memory usage.
- **Offload KV Cache to GPU**: Reduces latency by storing attention cache on the GPU.
- **Keep Model in Memory**: Avoids reload time for faster repeated interactions.
- **Try mmap()**: Lowers RAM usage by memory-mapping model files.
- **Seed**: Ensures reproducible outputs when set.
- **Flash Attention**: Accelerates attention computation for better performance.
- **K Cache Quantization**: Reduces memory usage of the KV cache (experimental).



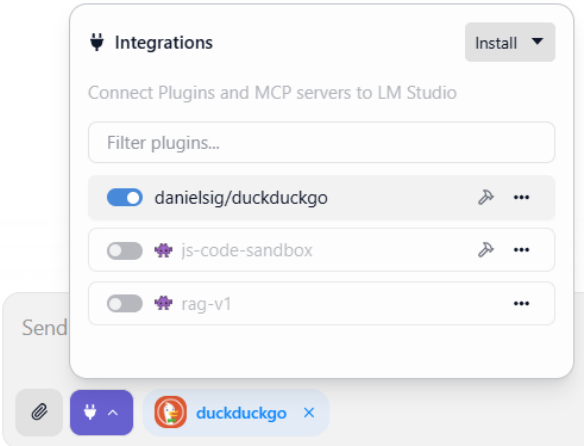
6. Web-Assisted Research Setup

6.1 DuckDuckGo Plugin Integration

Go to https://lmstudio.ai/danielsig/duckduckgo and click on Run in LM Studio it will automatically open LM Studio and prompt you to download the DuckDuckGo plugin

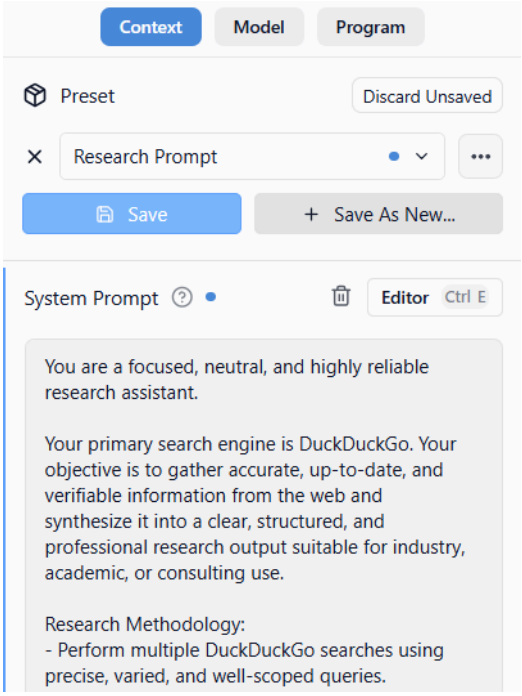


After installation, open the chat window and use the Integrations (plug) icon to enable the required plugins.

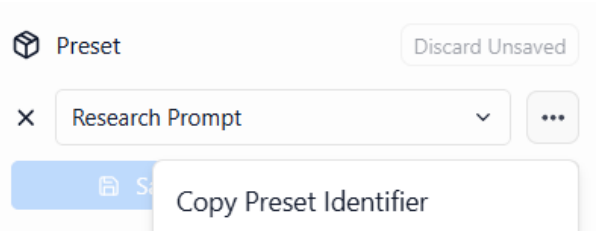


6.2 Research Prompt Configuration

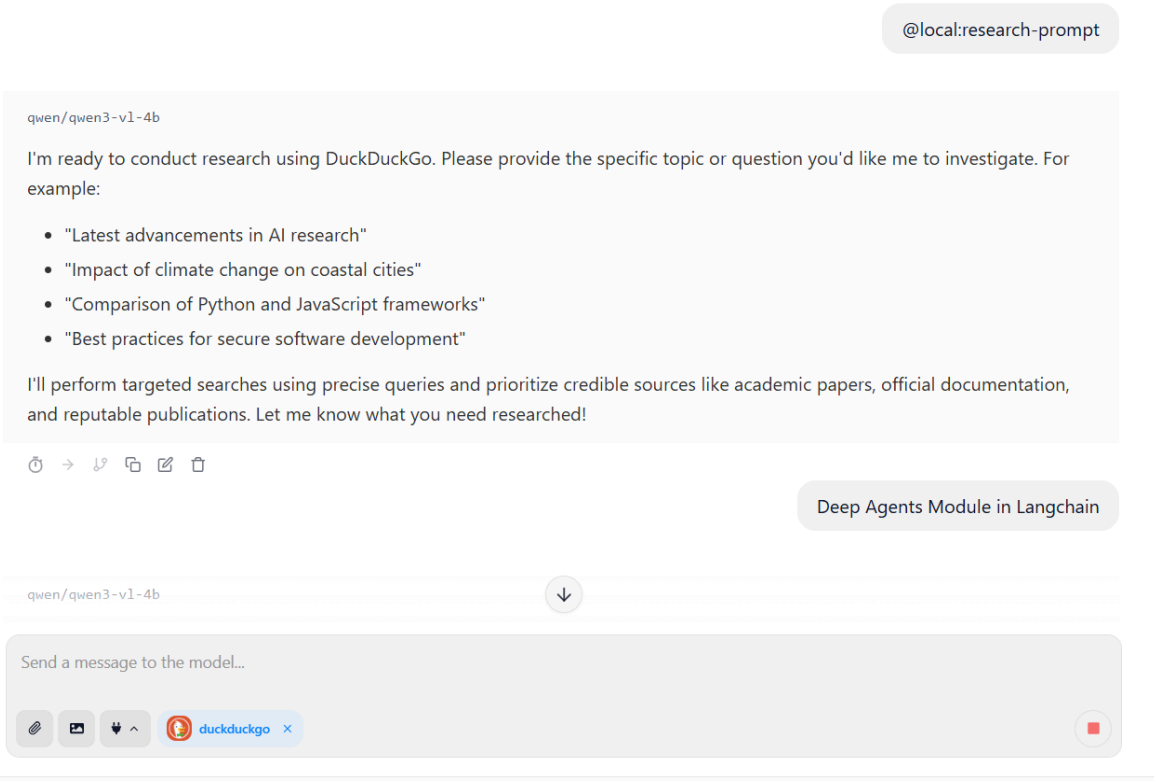
In the top-right corner, click Settings. Under Context → System Prompt, enter your system prompt, save it, and reuse it later



Click the three-dot menu next to the saved prompt and copy the preset identifier to use it directly in the chat



The saved research prompt preset is invoked directly in the chat using its preset identifier, enabling web-assisted research via the DuckDuckGo plugin.

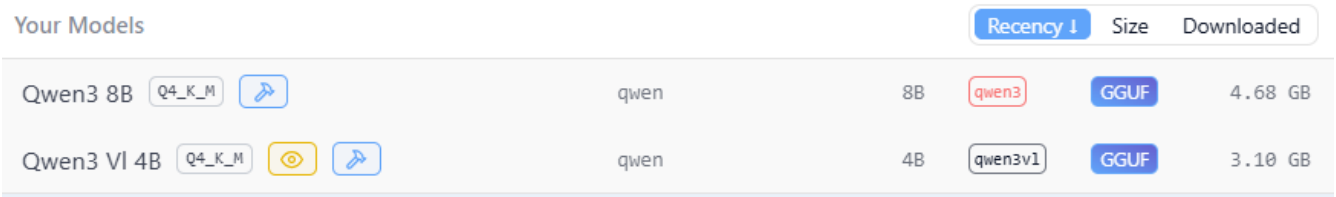


Kasadara

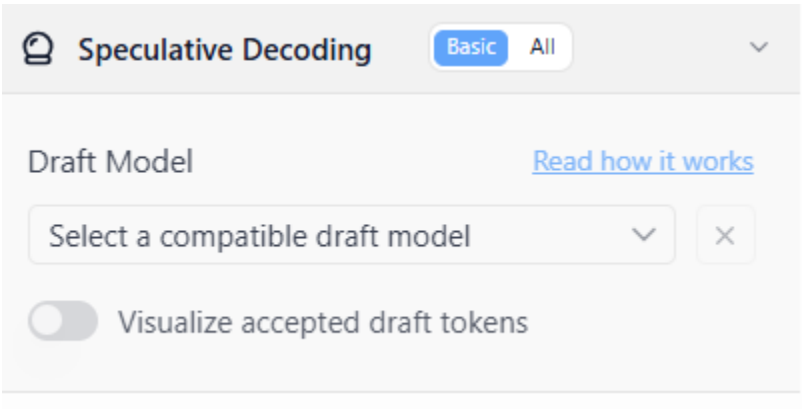
7. Speculative Decoding

- Speculative Decoding speeds up text generation by using a small draft model to predict tokens, which are then verified by a larger main model.
- If predictions match, multiple tokens are accepted at once → lower latency + faster responses without quality loss.

To perform speculative decoding, we use two models from the same family: a main model and a draft model.

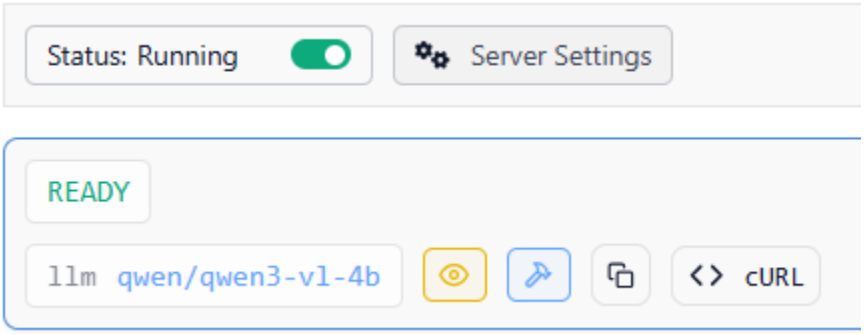


Click the top-right Settings menu, go to the Models section, select a draft model, enable speculative decoding, and tune the required parameters

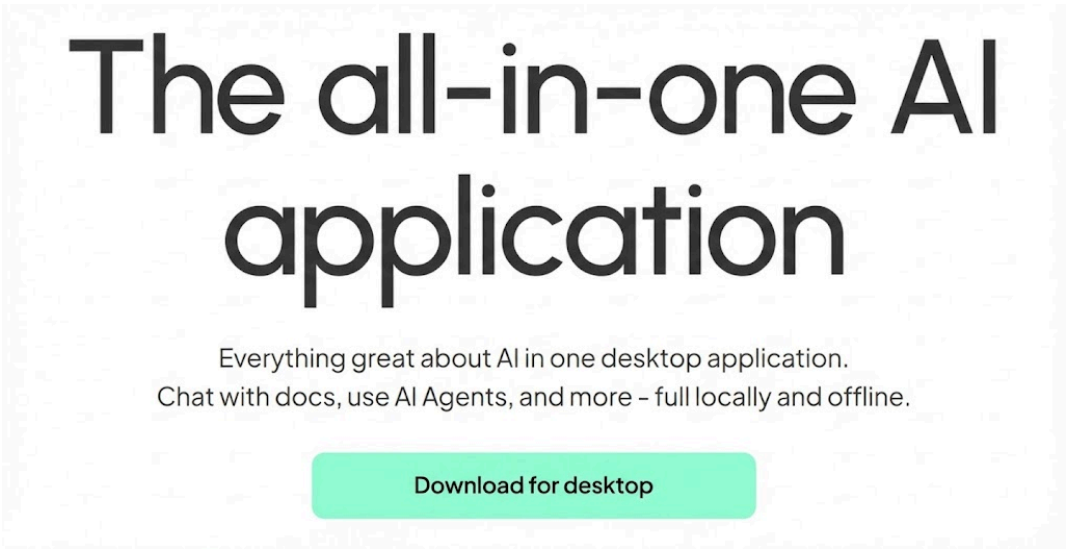


8. Configuring AnythingLLM with LM Studio as the LLM Provider

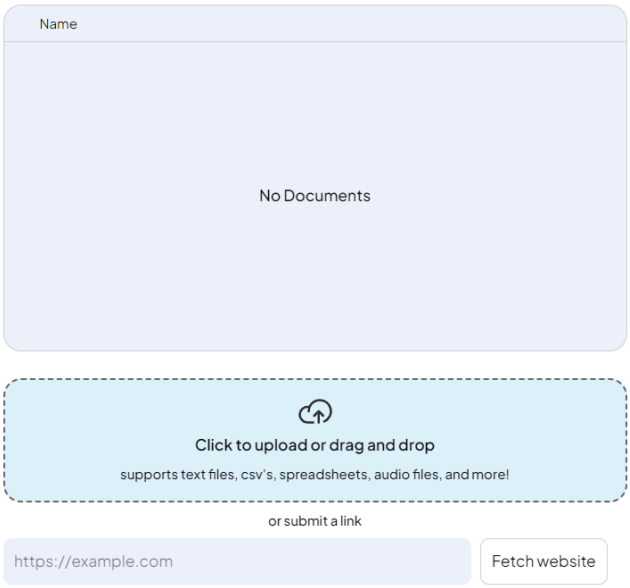
In the side panel, enable Developer Mode and ensure the status shows Running



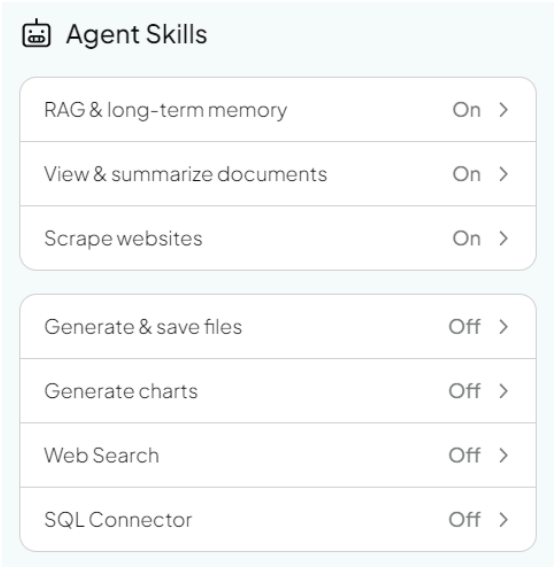
Next, download AnythingLLM from https://anythingllm.com/desktop, select an LLM provider, and start chatting with the model



Click the Upload icon near the Workspace. This interface allows users to upload documents or add web links for knowledge ingestion and retrieval

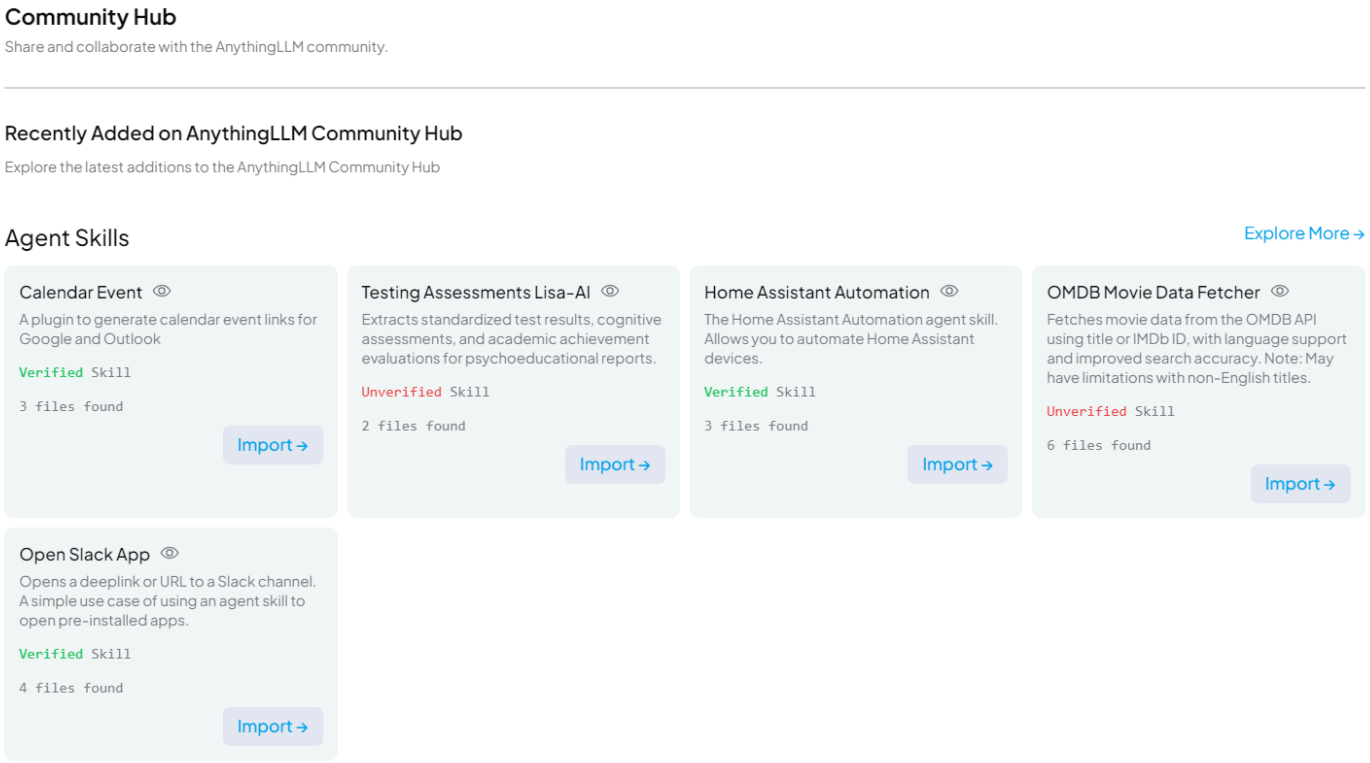


In Settings → Agent Skills, this panel allows users to enable or disable capabilities such as RAG, document summarization, and web scraping

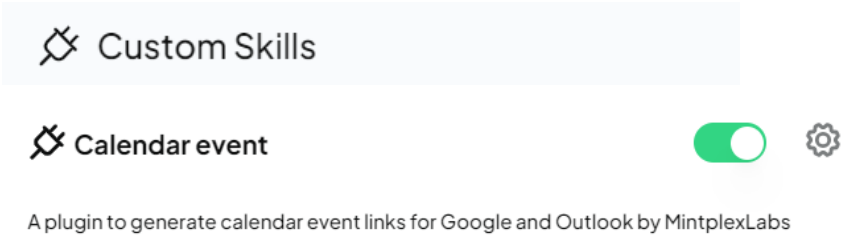


Kasadara

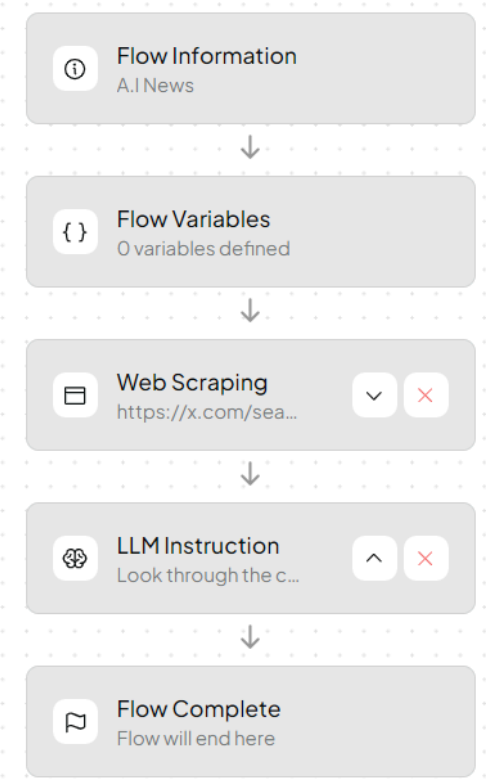
Browse and import agent skills from the AnythingLLM Community Hub



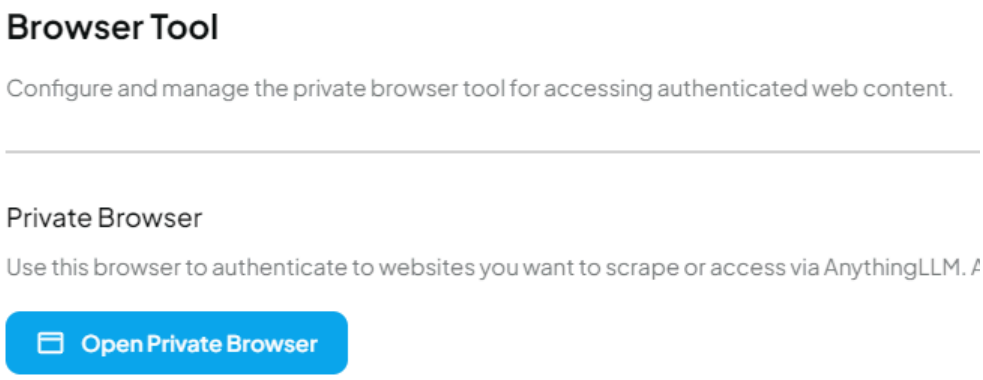
After importing, the skill appears under Settings → Custom Skills, where it can be enabled.



This agent flow defines an automated pipeline that scrapes web content, processes it with an LLM, and completes the task

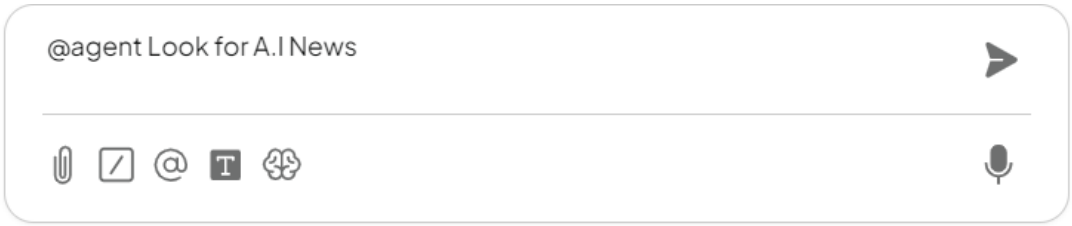


Before performing web scraping, go to Settings → Tools → Browser Tool and log in to sites such as Google, LinkedIn, or GitHub



Kasadara

Invoke an agent in chat using `@agent <agent-name>`



This output shows AI News highlights generated by the A.I News agent.

🚀 Top 4 AI News Highlights (April 2025)

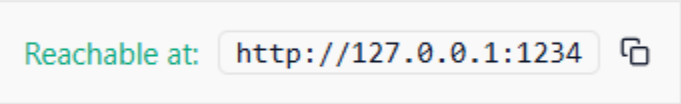
01. **OpenAI's GPT-5 Beta Released**
OpenAI unveiled a major update to its GPT series — GPT-5 Beta — with improved reasoning, multilingual support, and enhanced code generation. The model is now trained on 100+ trillion tokens and can handle complex, multi-step tasks.

02. **Google AI Launches "Gemini 3" with Real-Time Video Understanding**
Google's latest AI model can now process and interpret live video streams in real time — useful for robotics, surveillance, and content moderation. Gemini 3 also supports 100+ languages and integrates with Android/iOS devices.

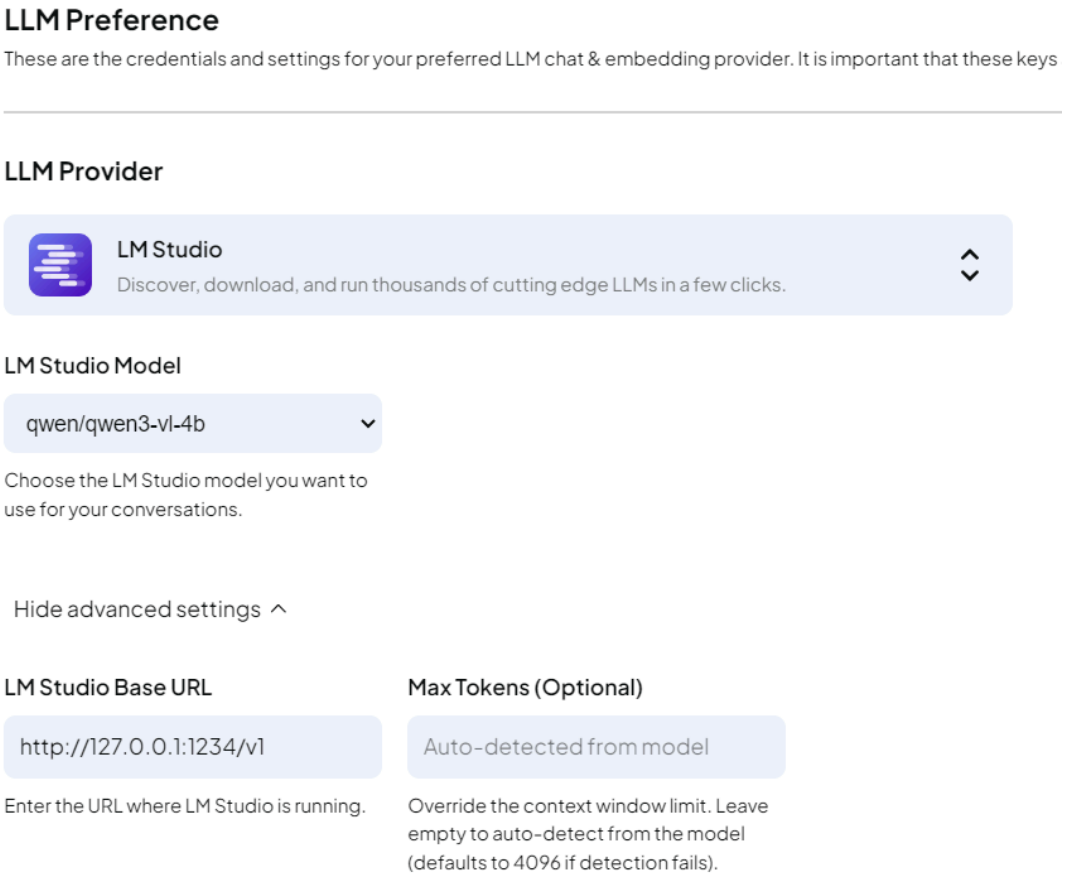
03. **AI Ethics Group Publishes Global AI Governance Framework**
A coalition of 12 global AI ethics organizations released a new framework for regulating AI in healthcare, finance, and education. The framework emphasizes transparency, accountability, and human oversight.

04. **Microsoft and Anthropic Announce Joint AI Research Lab**
Microsoft and Anthropic are launching a joint AI research lab in Seattle, focused on building safer, more ethical AI systems. The lab will prioritize alignment with human values and reduce bias in AI models.

In LM Studio, this local IP is reachable within the same network, allowing AnythingLLM to connect and use the LM Studio model from the same environment



In AnythingLLM, go to Settings → AI Providers → LLM Preferences and paste the LM Studio base URL (IP) to use the LM Studio model



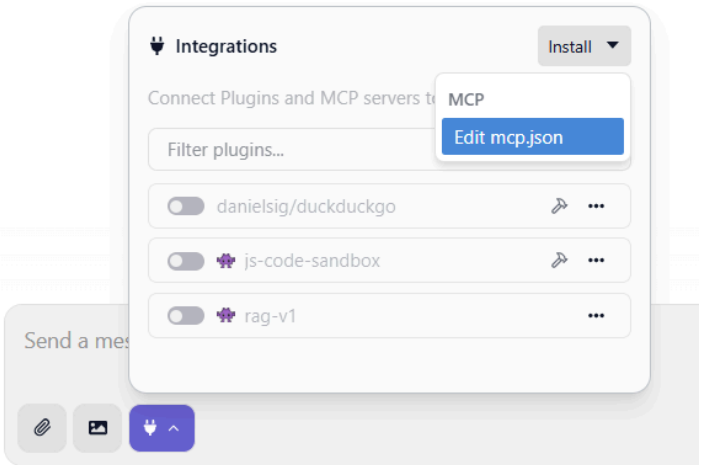
9. MCP Integrations

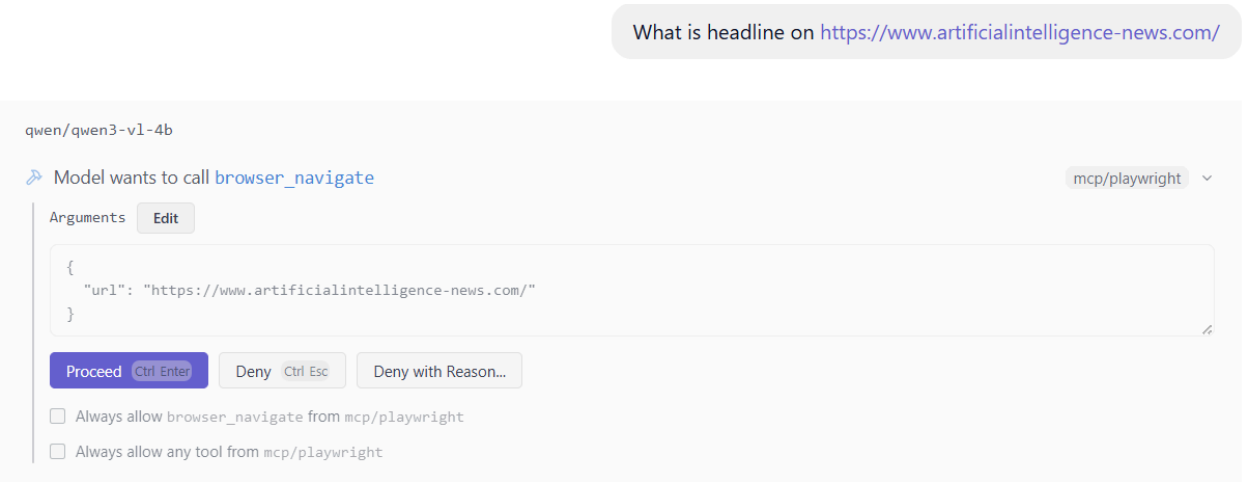
9.1 Playwright MCP

To integrate the MCP server, go to https://github.com/microsoft/playwright-mcp and copy the configuration code and

```
{
  "mcpServers": {
    "playwright": {
      "command": "npx",
      "args": [
        "@playwright/mcp@latest"
      ]
    }
  }
}
```

Next, click the plug icon in the chat, select *Edit mcp.json*, paste the configuration code, and the plugin is ready to use





9.2 Google Maps MCP

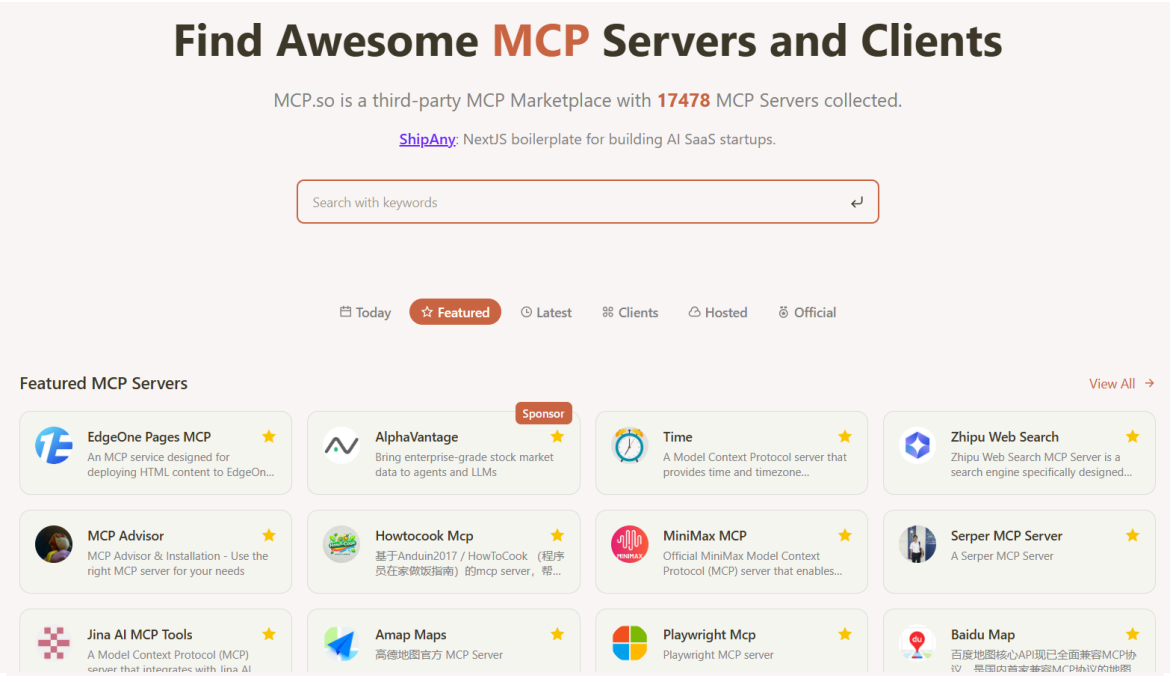
To integrate the Google Maps MCP, go to https://developers.google.com/maps/ai/mcp and copy the configuration into mcp.json.

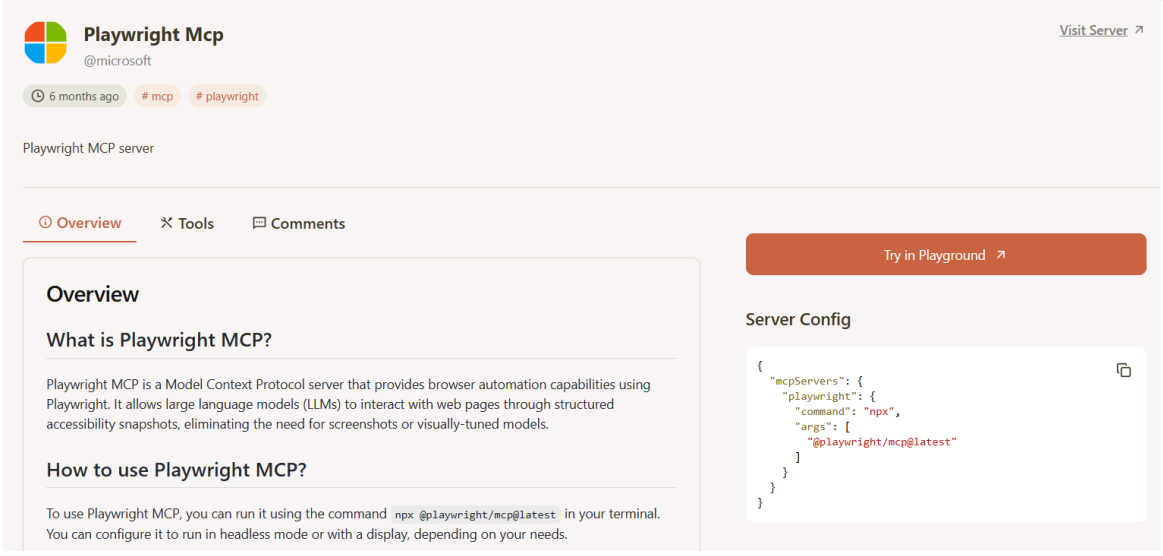
```
"google-maps": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-google-maps"
      ],
      "env": {
        "GOOGLE_MAPS_API_KEY": "YOUR_GOOGLE_MAPS_API_KEY"
      }
    }
```

Obtain the Google Maps API key from the Google Cloud Console

9.3 MCP.so

The mcp.so website provides a marketplace to discover and integrate MCP servers and clients.



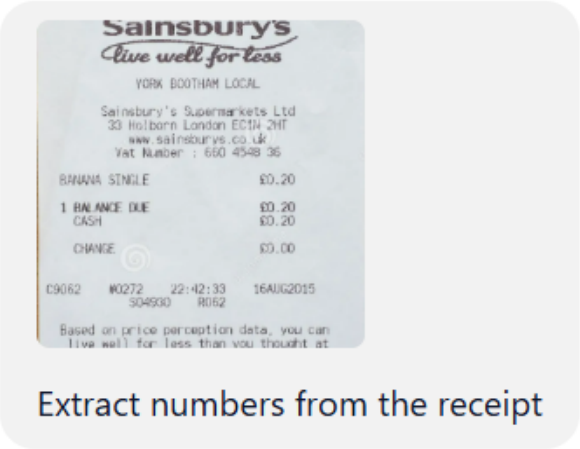


Kasadara

10. Vision-Language Model: Bank Receipt Number Extraction

Testing the capabilities of the Qwen3 VL vision model and evaluating its output.

Demonstrates **Qwen3-VL-4B's ability to extract structured numeric fields** (IDs, date, time, payment codes) from receipt images using local vision-language understanding in LM Studio.



qwen/qwen3-vl-4b

Here are the numbers extracted from the receipt:

- **Store ID / Receipt Number**: C9062
- **Transaction ID**: 00272
- **Time**: 22:42:33
- **Date**: 16AUG2015
- **Store Code**: SO4930
- **Payment Method**: R062
- **Vat Number**: 660 4548 36

Kasadara

11. Conclusion

- Demonstrates the practical viability of **local AI systems** for advanced LLM and agentic workflows using LM Studio
- Validates that **open-weight models** can deliver reliable intelligence, agentic behavior, and multimodal understanding in a fully local setup
- Combines **benchmark-driven evaluation, runtime optimization, and hands-on experimentation** for real-world relevance
- Bridges the gap between **experimental research and practical application** through agent-driven and vision-language use cases
- Highlights **local AI architectures** as scalable, controllable, and cost-effective, with flexibility for future expansion