Testing & Evaluation Sheet

Ollama

1. Tool Overview

Name:	Ollama
Category:	AI Chat
Purpose:	Run large language models locally on personal computers for private, offline AI assistance.
Date Tested	5/6/25
Status:	Deployed ☑ Operational - Actively running/maintained ☐ In Testing - Currently being evaluated or piloted ☐ Inactive/Deprecated - No longer maintained or functional ☐ CSOs - Verified adoption by one or more CSOs
Deployment Architecture:	☑ A standalone software - Runs entirely locally (e.g., runs on computer and doesn't depend on external server) ☑ A locally hosted service with separate server and client component - Run both backend/frontend yourself (e.g., backend could be on a local network, or self-hosted on cloud) ☐ A service with a local client that's hosted by a third party - You install a client on your device, but it connects to and depends on a remote server (e.g., Signal: install app (client), but Signal's servers handle message relaying, etc.) ☐ A service that is hosted by a third party but can also be self-hosted
Version:	v0.6.8

2. Installation & Setup

OS Compatibility	macOS (Intel and Apple Silicon), Windows, Linux (Ubuntu/Debian-based)
Installation Manual:	Yes: https://github.com/ollama/ollama

Installation Steps:	1. Visit https://ollama.com 2. Download the installer based on OS 3. Install: Run the installer and follow on-screen instructions 4. Model Setup: Use the command-line interface to pull desired models, e.g., ollama pull llama3 5. Run: Start the Ollama service and interact with models via CLI or integrated applications
Mention if command-line setup or special configurations are needed	Some: Exs. - For CLI: - curl -fsSL https://ollama.com/install.sh | sh - Run: - ollama run llama3
Common Installation Issues & Fixes:	“GPU not supported” - fix: use CPU-only flag “Model won’t start” - check RAM availability (some models need 8GB–16GB+) Firewall blocking download - download model manually from official repo
User Documentation:	Yes (https://ollama.com/library, https://github.com/ollama)
Required Technical Knowledge	Beginner to Intermediate - Basic familiarity with command-line interfaces is beneficial but not mandatory.

3. Testing & Evaluation

Category	Details	Score
Operational Functionality:	**Functionality** □ The tool is mostly non-functional with many broken features and bugs. □ Several broken features or bugs □ Minor bugs or issues □ Mostly functional with few bugs or no bugs ☑ Fully functional with no bugs - Executed core functions, including model loading, prompt interactions, and response generation. **Internet Dependence:** ● No internet needed post-installation (models run 100% offline) ● Low-Bandwidth Performance: Tested on 2G/3G networks; performance remains stable post-setup.	

	Localization & Language Support - Depends on the model being used, but most supports multiple languages, including English, Chinese, Japanese, Spanish, and Korean. - Community Contributions: Active community involvement in localization efforts. **Mobile Accessibility** - Mobile App: No dedicated mobile application available. - Mobile Browser Access: Accessible via mobile browsers; however, performance may vary based on device capabilities.	
Usability for Non-Technical Users	**Ease of Installation & Deployment** - Installation process is straightforward, with clear instructions provided - CLI-based, but 1-line command for install; ~10 min setup **User Onboarding Experience** - No in-app guidance; users rely on external documentation. **Technical Experience Level Required** - Intermediate for CLI, but improving	
Security & Privacy Strength	**Encryption Standards** - No specific encryption standards mentioned; relies on system-level security. - Not really applicable – runs offline, no data transmitted **Known Strength resilience** - Censorship Resistance: Operates offline, making it resilient to network-based censorship. - Ideal for offline/censored use; bypasses surveillance by avoiding web use **Comparison with Known Standards** - Large models consume heavy RAM/CPU - Excellent local privacy – better than ChatGPT or Gemini in secure environments - Aligns with best practices by minimizing data transmission and storage **Data Minimization** - No data is collected or transmitted; all operations are local **Privacy Policy Accessibility and Clarity**	

	● No formal privacy policy found; however, the tool’s local nature inherently supports user privacy	
Maintenance/Sustainability	**Community support** ● Active GitHub repository and community forums provide assistance and updates **Development active status** ● Update Frequency: Regular updates with recent version 0.6.8 released on May 3, 2025 (as of May 6th, 2025) ● Developer Responsiveness: Active engagement with community feedback and issue resolution. **Funding and Sponsorship** ● Community Driven ● Backed by Open Source contributors and private funders	
Performance / Effectiveness & Reliability	**Testing Environment Setup:** ● Device: Macbook Pro (14 inch, M4 Chip), 10-core CPU, 24 GB Memory ● OS: 15.2 Sequoia ● Network: Wifi **User Experience Observations** ● Smooth, fast for LLaMA 2/3 **Speed & Responsiveness:** ● Speed (Token Generation): ○ CPU (No GPU Acceleration): ~2000 ms/token ○ GPU (Consumer GPU): ~500 ms/token ● Inference Time: For a 10-token response: ○ CPU: ~20 seconds ○ GPU: ~5 seconds ● Efficient model loading and execution times. **Resource Usage:** ● 10–12 GB used when running LLaMA 3 8B ● Moderate usage; resource consumption varies based on the model size and complexity. **Network Performance:** ● None after initial model download **Reliability** ● 100% uptime in offline testing	
Deployment Considerations:	**Open Source & Transparency:** ● The source code is hosted on GitHub, allowing anyone to inspect, audit, or modify it.	

	● Core components and model-loading logic are openly maintained, although individual models pulled through Ollama (like Meta’s LLaMA) may have separate licenses or restrictions. **Cloud vs. Local Deployment:** ● Entirely local. No cloud infrastructure needed **Dependencies:** ● None required beyond binary. Optional: Docker, Make, etc. for dev builds. **Post-Deployment Maintenance** ● Maintenance: ○ It runs locally with no external dependencies once installed. ○ Updates (e.g., new model versions or bug fixes) are applied by pulling the latest version via GitHub or reinstalling via their install script. ○ Logs are local; no backend server to maintain unless the user explicitly builds a web UI or integrates Ollama into larger systems. ● Monitoring: ○ Minimal required—main concern is available system resources (RAM/GPU) and occasional compatibility checks after OS updates. ● Forking: ○ Straightforward using GitHub. The project supports community contributions via pull requests. ○ Configuration for models and system behaviors is managed via CLI and can be extended by editing config files or the command logic. **Merge/Sustainability:** ● GitHub repository includes build and contribution instructions. ● Issues and discussions show active developer responses, so those forking the project can get community help.	

4. Testing Scenarios

● **Scenario 1**	● Use Case: Running LLaMA 3 8B to summarize long PDFs using external integration ○ Result: Successful. Summary generated in ~30s via CPU (tested with ollama-python) ○ Notes: Integration with LangChain, ollama-python, or LM Studio works well for pipelines.
● **Scenario 2**	● Use Case: isolated environment test – no internet access after installation ● Result: 100% successful ● Notes: Confirmed no external calls; Ollama runs entirely offline, ideal for censored environments.
5. Insights & Recommendations	
Key Findings	**Strengths:** ● Full offline functionality ● Beginner-friendly CLI ● Fast, local LLM performance ● Privacy-preserving ● Cross-platform support ● Active and helpful open-source community **Weaknesses:** ● No GUI by default (CLI only) ● No mobile app yet ● Large models need powerful hardware ● Limited error messaging if setup fails
Suggested Improvements	- Official GUI app - Better in-app onboarding or usage guidance - Windows installer could improve model pull error handling
Alternative Tools:	- LM Studio: Offers GUI for local LLMs using Ollama backend - GPT4All: Also local models, slightly less polished - LocalAI: Fully open-source but more complex setup - Open WebUI: Local LLM frontends integrating Ollama
License	- MIT License (Ollama is open-source: GitHub repo)
Cost/Resource Implications	**Total Cost of Ownership:** ● Free to Use, no premium tiers ● Hardware Cost: Needs 8–16 GB RAM for 8B models ● Maintenance Cost: Minimal – updates handled via CLI ● Hidden Costs: None; fully transparent open-source tool

Why is this useful to civil societies in authoritarian environments?	Ollama provides a unique advantage to civil society organizations operating under authoritarian regimes due to the following factors: - Total Offline Capability: After installation, all AI inference occurs locally, without requiring an internet connection. This ensures operability even in blackout zones or surveillance-heavy environments. - Censorship Resistance: The tool does not rely on DNS, APIs, or external cloud services, making it immune to common censorship methods such as IP blocking or DPI (deep packet inspection). - Privacy and Anonymity by Design: No data is transmitted, stored remotely, or collected, making it safer than centralized AI platforms (e.g., ChatGPT, Bard) that require persistent cloud connections. - Open-Source and Modifiable: CSOs can inspect, customize, or self-host components of Ollama to suit their regional needs or integrate with other privacy-preserving tools. - Resilience for Fieldwork: Can be installed on laptops used in isolated regions or disaster recovery areas without any dependency on Western infrastructure. - Empowers Local Capacity-Building: Activists and developers can locally fine-tune or extend models for tasks like translation, legal analysis, or media fact-checking—without needing access to foreign cloud tools. This positions Ollama as a critical infrastructure tool for digital sovereignty, especially valuable for: - Journalists avoiding surveillance - Legal teams working offline - NGOs conducting fieldwork in censored areas - Human rights monitors and whistleblowers