ChatGPT

PRD-Update: Local AI Chat Enhancements

Based on survey of open-source Ollama clients (Askimo, LLocal, Ollamate), we propose the following feature enhancements. Each proposal is assessed against the existing guardrails (message immutability, intent supremacy, backend-only model access, append-only logs [1] [2]) and marked as either **allowed** or **requiring PRD deviation**.

- **Virtualized chat transcript with lazy history loading (Allowed):** Improve UI performance for long conversations by virtualizing the message list. Only a window of recent messages is rendered, and older messages are fetched/rendered on-demand when the user scrolls up [3]. This preserves append-only logs (no editing or deletion) and does not alter backend logic or message integrity [1]. Askimo's implementation explicitly uses this approach to keep memory usage low [3].

- **Searchable conversation history and exportable logs (Allowed):** Enable full-text search over past messages and allow exporting chat transcripts (e.g. to Markdown, JSON, or HTML). Askimo supports in-chat search of any message, star/pin conversations, and one-click export of chats [4]. LLocal similarly indexes local chats for retrieval. These features only read from the immutable SQLite log and produce new output files; they do not violate the append-only, no-silent-change rule. (Exported logs can aid backup or sharing without altering stored state.)

- **Theming and keyboard shortcuts (Allowed):** Support multiple UI themes (light/dark and user-selected color schemes) and comprehensive keyboard navigation. LLocal already offers several themes including dark mode [5]. Askimo advertises "custom themes" and keyboard hotkeys for all actions [6]. These are purely presentation enhancements and do not affect chat logic or storage, fully respecting guardrails.

- **Runtime model/provider switching (Allowed):** Allow the user to switch between local models (via Ollama) or cloud providers at runtime via configuration settings. Askimo provides one-click switching between local Ollama models and OpenAI/Gemini/etc. [4]. In our architecture, the UI would send chat requests to the backend, which consults a model selection setting. This follows the implementation plan (backend-only model calls) and requires no deviation: changing the active model is purely a configuration change, not user-message content.

- **Full-text search and file-based RAG indexing (Allowed):** Index user-supplied files (PDF, DOCX, etc.) into a local retrieval-augmented-generation (RAG) system so the chat can answer questions about those documents. Askimo calls this "project-aware RAG" [7]; LLocal likewise "persist[es] files in a local vector DB" for Q&A [8]. As long as file ingestion is initiated by the user (an explicit action) and treated as static context in the conversation, this does not violate user intent or add hidden state. (All data remains local and append-only: the vector store is a form of user-provided knowledge base.)

- **Embedding artifact outputs from tool runs (Requires PRD deviation):** Support inserting rich outputs (e.g. images, charts, or formatted tables) produced by auxiliary tools or code executions into the chat. LLocal hints at advanced "tool" agents and `<think/>` code blocks, but the current PRD

does not define non-text message types or integrated visual outputs. Allowing tool-run artifacts would require extending the PRD to clarify how such content is represented and stored (ensuring, for example, each artifact is tied to a specific user message and treated as immutable). This is a useful feature but beyond the original spec, so it **requires a formal PRD deviation request** to design safely without breaking append-only/message-integrity rules.

All of the above proposals are designed to preserve the core invariants: user messages remain immutable and append-only [1], no new silent state changes are introduced, and all model/tool invocations continue via the backend [2]. In summary, features like virtualization, search, export, theming, and model-switching fit squarely within the guardrails [4] [3]. Only the richer *artifact embedding* functionality would need careful specification (and thus is flagged as a deviation).

Sources: Askimo feature descriptions [4] [3] and LLocal README [8] [5], with guardrail summaries from project docs [1] [2].

---

[1] [2] guardrails.md
file://file-UGRkvxtwUojw2UzWTC6vjs

[3] [4] [6] [7] Askimo: Ollama Desktop App & GUI for Llama 3, Mistral & Local AI Models (2025) | Askimo
https://askimo.chat/blog/askimo-with-ollama-the-best-desktop-for-local-ai/

[5] [8] GitHub - kartikm7/llocal: Aiming to provide a seamless and privacy driven chatting experience with open-sourced technologies(Ollama), particularly open sourced LLM's(eg. Llama3, Phi-3, Mistral). Focused on ease of use. Available on both Windows and Mac.
https://github.com/kartikm7/llocal