Feature Prioritization Matrix — MoSCoW Framework

This matrix aligns development efforts with user value and technical constraints, especially for a local-first AI assistant.

---

1) M — Must Have

Core features required for a functional, independent offline AI system

Feature	Reasoning	User Impact
Local LLM Response Generation	Primary product capability	Enables essential writing and Q&A
Offline Operation (No Cloud Dependency)	Addresses outage problem directly	Reliability in restricted networks
Local Document Ingestion (PDF Upload)	Foundation of personalized knowledge	Retains contextually relevant answers
RAG Retrieval Pipeline (ChromaDB)	Accurate info from user memory	Trustworthiness of output
Safety + Error Handling	Prevent crashes during critical use	Smooth user experience

📝 These define JoelGPT's **minimum viable product**.

---

2) S — Should Have

Enhancements improving experience & scalability

Feature	**Benefit**	**User Segment**
Web Search Command (/search)	Access external updates when needed	Professionals, job seekers
Real-Time Streaming UX	Reduces response wait frustration	All
Duplicate Document Check	Saves storage and avoids confusion	Heavy RAG users
Basic Prompt Context Guardrails	Ensures model does not hallucinate	Academic + Technical users

📝 These unlock hybrid *offline + controlled online augmentation*.

---

3) C — Could Have

High-value, future-scope improvements if time/resources allow

Feature	**Value**	**Notes**
Multi-format ingestion: DOCX, TXT, HTML	Expands knowledge base support	Moderate effort
Encrypted Vector Storage	Increased trust and enterprise use	Requires security library
UI Dashboard for Document Management	Better accessibility	React/Flask integration
Configurable Retrieval Settings	More control on RAG outputs	Advanced filtering
Version Control of Uploaded Knowledge	Track material changes	Supports long-term use

📝 These features enhance usability and privacy polish.

4) W — Won’t Have (Now)

Nice ideas, but do not align with current constraints

Feature	Reason	Misalignment
Competing with cloud SOTA models	Beyond hardware limits	Not the product objective
Heavy GPU dependencies	Limits adoption for students	Opposes “low compute” goal
Advanced multimodal input (video/audio)	Resource-heavy	Future roadmap only
Complex conversational memory	Model size limitations	Risk of degraded performance

📝 These could come later with hardware upgrades or bigger models.

5) Resulting Priority Order (Summary)

Priority Level	Strategy
Must Haves	Core problem-solution fit
Should Haves	Competitive differentiators
Could Haves	Experience uplift
Won’t Have	Keeps project lean & focused

6) Why This Prioritization Works

Consideration	**Impact**
Reliability under outages	Highest priority
Limited hardware	Constraints model choices
Privacy expectation	Drives local-only processing
Student & early-professional target	Cost-friendly approach
Avoiding unnecessary complexity	Faster delivery and stability