Multimodal RAG System - Offline Mode Masterplan

1. App Overview and Objectives

Project: Multimodal Retrieval-Augmented Generation (RAG) System — Offline Mode
Objective: Build a fully offline system that ingests, indexes, and queries diverse data formats (PDF, DOCX, images, audio) in a unified semantic retrieval framework, providing **grounded LLM-generated responses with citations**.
Goals for Hackathon Demo:
- Showcase cross-modal retrieval and LLM-synthesized answers
- Fully offline operation with persistent vector indices
- Clear, functional interface with inline previews and citations

2. Target Audience

- Analysts, researchers, and officers at NTRO
- Users who handle diverse multimodal documents and need **fast, evidence-backed retrieval**
- Hackathon judges (interface clarity secondary to functional offline retrieval and citation transparency)

3. Core Features and Functionality

Hackathon Must-Haves

- **Batch ingestion:** Upload PDF, DOCX, images, audio at once; preprocess with OCR/STT/embeddings
- **Offline vector indexing:** Unified vector space stored locally (FAISS or equivalent)
- **Top-k semantic retrieval:** Cross-modal search returning text, image, and audio results
- **LLM RAG integration:** Generate concise, context-aware answers from retrieved items
- **Inline previews:** PDF snippets, image thumbnails with OCR text, short audio snippets
- **Citations:** Numbered references linking to original files, pages, timestamps, or image regions

Nice-to-Haves (Time Permitting)

- Query history panel
- Export retrieved snippets (PDF/CSV)
- Batch report generation
- Drag-and-drop uploads
- Minor image preprocessing (resize, contrast adjustment)
- Keyword highlighting in text snippets
- Modality filters (Text/Image/Audio)

4. High-Level Technical Stack Recommendations

Component	Recommendation (Offline-Friendly)	Rationale
Text Embeddings	MiniLM / MPNet variant	Compact, efficient semantic embeddings
Image Embeddings	CLIP-small + OCR via Tesseract	OCR for text, CLIP for semantic similarity
Audio STT & Embeddings	Whisper-small / Vosk + embeddings	Fully offline, compact, supports semantic retrieval
LLM	LLaMA 2 7B (quantized)	Offline inference with sufficient context for RAG
Vector Index	FAISS (local, persisted)	Fast similarity search across modalities
Preprocessing	Minimal modular pipelines (metadata extraction, normalization)	Keeps system efficient and modular
UI Framework	Lightweight GUI (Python: Tkinter/Qt, Web: Electron/React local)	Focus on clarity and inline previews

5. Conceptual Data Model & Processing Pipeline

Data Flow:

```
[User Query / File Upload]
        ↓
[Preprocessing]
  Text: full text, headings, tables, metadata
  Images: OCR + CLIP embeddings
  Audio: Whisper STT + embeddings
        ↓
[Vector Index / FAISS] → persisted locally
        ↓
[Retrieval] → top-k cross-modal results
        ↓
[LLM RAG Integration] → grounded answer
        ↓
[UI Display] → ranked results, inline previews, citations
```

Unified Vector Space:
- Stores embeddings for all modalities in a shared space
- Enables cross-modal retrieval without separate mapping layers

6. User Interface Design Principles

- **Panels:** Left (uploads), center (chat/search), right (results)
- **Results display:** All-in-one ranked list with inline previews
- **Interactions:** Expandable citations, copy text, play audio inline
- **Optional filters:** Modality buttons (Text/Image/Audio)

7. Security and Privacy Considerations

- Fully **offline** — no network calls at runtime
- All uploaded files remain **local to laptop**
- Optional **privacy simulation**: masking or limiting displayed snippets
- Supports responsible handling of sensitive content

8. Development Phases / Milestones

Phase	Milestone
Phase 1: Core Pipeline	Batch ingestion, preprocessing (OCR/STT/embeddings)
Phase 2: Indexing	Unified vector space, FAISS persistence
Phase 3: Retrieval + LLM	Top-k retrieval, LLM integration for grounded answers
Phase 4: UI & UX	Inline previews, citations, audio playback
Phase 5: Demo Polish	Optional filters, minor preprocessing, export features

9. Potential Challenges & Solutions

- **Offline LLM inference:** Use small/quantized LLMs; limit top-k context
- **Resource constraints on laptop:** Optimize embeddings, batch preprocessing, minimize memory footprint
- **Cross-modal retrieval quality:** Use unified vector space, modular preprocessing (OCR, STT, normalization)
- **Citation precision:** Store metadata (page, paragraph, timestamp, image region) during preprocessing

10. Future Expansion Possibilities

- Support **larger datasets** (hundreds/thousands of documents/images/audio)
- Multi-user or shared deployments with offline syncing
- Additional modalities: video, advanced summarization, cross-document reasoning
- Model upgrades: swap in larger LLMs or embedding models
- Enhanced UX: query history, batch report generation, drag-and-drop uploads