Paperless-GPT

Custom AI Integration for paperless-ngx Document Management

Project Overview

Built custom AI services that extend **paperless-ngx**, an open-source document management system. My integrations add intelligent OCR, automatic classification, and semantic search capabilities that the base platform doesn't provide.

What I Built vs. What I Used:
Base Platform: paperless-ngx (open-source, not my code)
My Custom Services: paperless-gpt (Go) + paperless-chroma (Python)

Key Achievement: Natural language queries like "What were last month's expenses?" return relevant documents instantly using semantic similarity rather than keyword matching.

Architecture

```
Document Upload → Text Extraction → Embedding Generation ↓ ChromaDB
(Vector Store) ↓ User Query → Query Embedding → Similarity Search →
Context ↓ LLM (GPT-4/Ollama) ↓ Natural Language Response
```

Core Features

Document Processing

- PDF, image, and text document support
- OCR for scanned documents
- Automatic text extraction
- Metadata extraction (dates, amounts, entities)

Vector Search

- Sentence transformer embeddings
- ChromaDB for vector storage
- Semantic similarity search
- Hybrid search (vector + keyword)

LLM Integration

- OpenAI API (GPT-4) support
- Local models via Ollama
- Context-aware responses
- Source document citations

Technical Implementation

Embedding Pipeline

- Sentence Transformers for text embeddings
- Chunking strategy for long documents
- Overlap handling for context preservation
- Batch processing for efficiency

Query Processing

- Query embedding generation

- Top-K similarity retrieval
- Re-ranking for relevance
- Context window management

Multi-Provider Support

- Abstracted LLM interface
- OpenAI, Anthropic, Google support
- Local model fallback (Ollama)
- Cost optimization with model selection

Tech Stack

> **Language:** Python
> **Vector DB:** ChromaDB
> **Embeddings:** Sentence Transformers
> **LLM:** OpenAI API, Ollama
> **API:** FastAPI/Flask
> **Deployment:** Docker

Skills Demonstrated

Python ChromaDB Vector Databases OpenAI API LLM Integration Semantic Search Document Processing REST API Docker

Deliverables

- Vector database for semantic document search
- LLM integration for natural language queries
- Automatic document classification

- REST API for document operations
- Docker deployment configuration
- Multi-provider LLM support
- Source citation in responses

Why This Matters

This project demonstrates how to extend existing open-source tools with custom AI capabilities. Rather than building from scratch, I integrated modern AI/ML techniques (vector search, LLMs) into an established platform. The RAG pattern used here is applicable to any system requiring intelligent document search.