

# MÔ TẢ ĐỒ ÁN CUỐI KỲ

## Tên đề tài:

Ứng dụng Chat AI chạy Local LLM tích hợp Model Context Protocol (MCP) để tìm kiếm và phân loại nội dung file văn bản.

## I. PHẦN CƠ BẢN

### 1. Mục tiêu:

Xây dựng ứng dụng chat offline AI có khả năng:

- Nhận câu hỏi qua giao diện chat
- Tìm kiếm nội dung trong thư mục file (PDF, Word, PPT)
- Phân loại file dựa trên nội dung
- Gửi metadata file (tên, nhãn, nội dung) qua API MCP Cloud
- Hiển thị kết quả trên giao diện chat

Chạy offline, không Internet, không cần GPU.

### 2. Chức năng chính (ví dụ minh họa):

STT	Chức năng	Ví dụ minh họa
1	Giao diện Chat	User: "Tìm file có 'kế hoạch 2024'"
2	Local LLM xử lý prompt	AI: "Đang tìm kiếm trong thư mục..."
3	MCP Filesystem index	Tìm thấy: plan2024.pdf, marketing2024.pptx
4	AI phân loại file	plan2024.pdf → Nhóm A; marketing2024.pptx → Nhóm B
5	MCP Cloud gửi metadata	{"filename": "plan2024.pdf", "label": "A"}
6	Hiển thị kết quả	AI: "Đã tìm thấy 2 file, metadata đã gửi."

### 3. Cấu hình:

- Chạy trên CPU: llama.cpp hoặc Ollama (quantized model)
- Không cần GPU
- Offline 100%
- Máy tính phổ thông (RAM 8-16GB)

## II. PHẦN NÂNG CAO

### 1. Agentic AI – Tự động lên kế hoạch hành động:


Ví dụ:

User: “Tìm file ‘kế hoạch marketing 2024’ và xuất danh sách.”

AI tự thực hiện:

1. Tìm file → index nội dung → phân loại → xuất metadata

2. Trả kết quả:

 Đã tìm thấy:


- marketing\_plan.pdf → Nhóm A

- sales\_strategy.pdf → Nhóm B

Đã xuất file metadata.xlsx

### 2. Chain of Thought (CoT) – AI giải thích quy trình:

Ví dụ kết quả:

 Tôi đã thực hiện:

- Bước 1: Index 15 file PDF, Word

- Bước 2: Tìm thấy 2 file chứa ‘kế hoạch marketing 2024’

- Bước 3: Phân loại:

+ marketing\_plan.pdf: Nhóm A

+ sales\_strategy.pdf: Nhóm B (liên quan gián tiếp)

### 3. RLHF – Học từ phản hồi người dùng:

Ví dụ:

AI: Tôi gán marketing\_plan.pdf → Nhóm A. Bạn đồng ý không?

User: Không, nên là Nhóm B.

AI: Đã cập nhật thành Nhóm B.

## III. Ý NGHĨA PHẦN NÂNG CAO

Tính năng	Ý nghĩa thực tiễn
Agentic AI	AI tự động hóa, giảm thao tác thủ công
Chain of Thought	Giúp user hiểu quy trình AI xử lý → tăng trust
RLHF	AI dần thích ứng theo phản hồi người dùng

## IV. SO SÁNH PHẦN CƠ BẢN & NÂNG CAO

Nội dung	Phần cơ bản	Phần nâng cao
----------	-------------	---------------

AI xử lý	Theo prompt	Tự plan step
Hiển thị	Danh sách file	Kèm giải thích CoT
Người dùng chỉnh	Không	Có (RLHF)
AI thích ứng	Không	Có

### V. KẾT QUẢ MONG ĐỢI

- Ứng dụng chat offline, AI chạy local LLM, không GPU
- Tìm kiếm + phân loại file
- Tích hợp MCP (Filesystem + API Cloud)
- (Nâng cao) Có CoT + RLHF
- Video demo + kiến trúc + mã nguồn

### VI. CÁC CÔNG NGHỆ ÁP DỤNG

Nhóm công nghệ	Công cụ / Framework / Mô tả
Local LLM	llama.cpp, Ollama, GGUF quantized models – Chạy mô hình LLM trên CPU
Model Context Protocol (MCP)	Giao thức tích hợp AI với Filesystem & API Cloud
Natural Language Processing	Xử lý ngôn ngữ tự nhiên qua Local LLM
File Indexing	PyPDF2, python-docx, python-pptx – Đọc/trích xuất nội dung file
Metadata API	FastAPI hoặc Flask – triển khai API MCP Cloud nhận metadata
Giao diện Chat	Tkinter, PyQt, Gradio, Streamlit
Data Storage	SQLite, JSON, Excel (pandas, openpyxl để export metadata)
Agentic AI (nâng cao)	langchain, crewAI hoặc lập kế hoạch tay
Chain of Thought	Triển khai CoT qua logic xử lý output → text giải thích từng bước
RLHF cơ bản (offline)	Ghi log phản hồi user vào JSON/SQLite → áp dụng rule lần sau
Triển khai offline	Toàn bộ ứng dụng chạy local, không internet, phù hợp máy RAM 8GB+