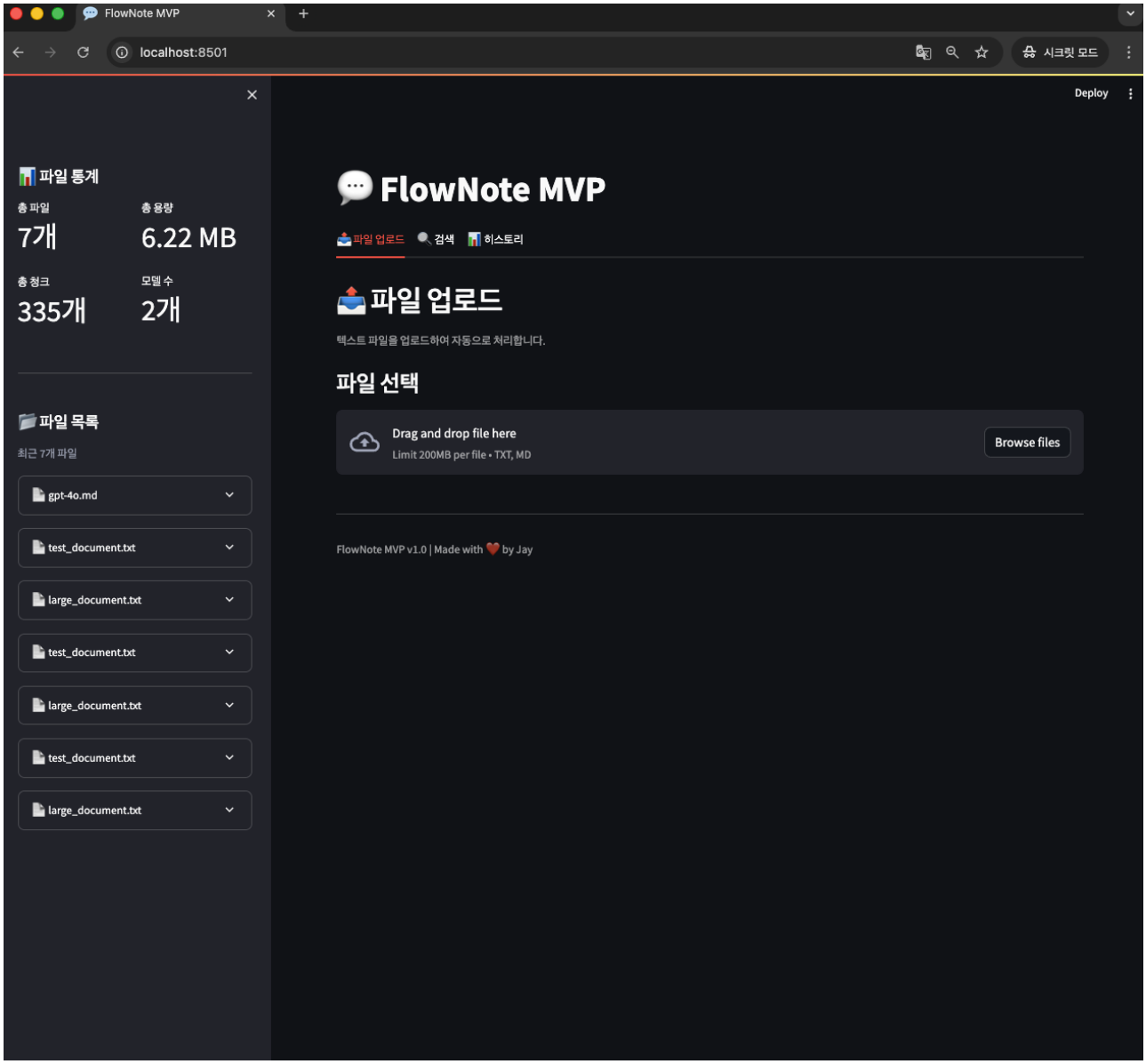
📚 FlowNote MVP

> AI 대화를 체계적으로 저장하고, 필요할 때 빠르게 찾아 활용하는 로컬 우선 문서 검색 도구

📖 프로젝트 소개

FlowNote는 ChatGPT, Claude, Perplexity 등 AI 도구와 나눈 소중한 대화들을 **흘려보내지 않고** 체계적으로 저장하고, 나중에 쉽게 찾아볼 수 있게 도와주는 **로컬 우선(Local-First)** 문서 관리 도구입니다.

- 

💡 왜 FlowNote인가?

- 🤔 **"저번에 AI한테 뭐라고 물어봤더라?"** - 기억이 안 나는 경험, 누구나 있죠?
- 📝 **"그때 받은 답변이 정말 좋았는데..."** - 다시 찾으려니 막막하셨나요?
- 💼 **"프로젝트 자료로 쓰고 싶은데..."** - 체계적으로 정리되어 있지 않아 어려우셨나요?

FlowNote는 이런 고민을 해결합니다!

✨ 핵심 기능 (v1.0 MVP)

📤 파일 업로드 & 처리

- **다양한 파일 형식 지원**: `.md`, `.txt`, `.pdf` 파일 업로드 가능
- **자동 청킹**: RecursiveCharacterTextSplitter로 적절한 크기로 분할
- **임베딩 생성**: OpenAI `text-embedding-3-small` 모델 사용
- **메타데이터 관리**: 파일명, 크기, 청크 수, 업로드 시간 자동 기록

🔍 의미 기반 검색

- **FAISS 벡터 검색**: 키워드가 아닌 **의미**를 이해하는 검색
- **유사도 점수**: 각 검색 결과의 관련도를 0-100% 점수로 표시
- **검색 히스토리**: 이전 검색 내역을 자동 저장하고 재사용 가능
- **빠른 응답 속도**: 평균 0.3초 이하의 빠른 검색

📊 대시보드 & 통계

- **파일 통계**: 전체 파일 수, 총 청크 수, 평균 유사도 한눈에 확인
- **파일 목록**: 최근 업로드한 파일 10개를 사이드바에서 확인
- **검색 히스토리**: 최근 검색 5개를 빠르게 재실행

🎨 직관적인 UI/UX

- **Streamlit 기반**: 간결하고 사용하기 쉬운 웹 인터페이스
- **실시간 피드백**: 파일 업로드, 검색 진행 상황을 실시간으로 표시
- **반응형 레이아웃**: 사이드바와 메인 영역으로 깔끔하게 구성

🚀 빠른 시작

1️⃣ 사전 요구사항

- **Python**: 3.11.10 이상
- **pyenv**: Python 버전 관리 (권장)
- **OpenAI API Key**: OpenAI Platform에서 발급

2️⃣ 설치 방법

```
# 1. 리포지토리 클론
```

```
git clone https://github.com/jjaayy2222/flownote-mvp.git
cd flownote-mvp

# 2. Python 버전 확인 (자동)
# .python-version 파일이 있어서 자동으로 3.11.10 사용

# 3. 가상환경 생성 및 활성화
# pyenv virtualenv 3.11.10 <원하는 가상환경명>
pyenv virtualenv 3.11.10 myenv
pyenv activate myenv

# 4. 패키지 설치
pip install -r requirements.txt

# 5. 환경변수 설정
cp .env.example .env
# .env 파일을 열어 OPENAI_API_KEY 입력
# 파일명을 .env로 복사해서 다시 설정하기
```

3️⃣ 실행

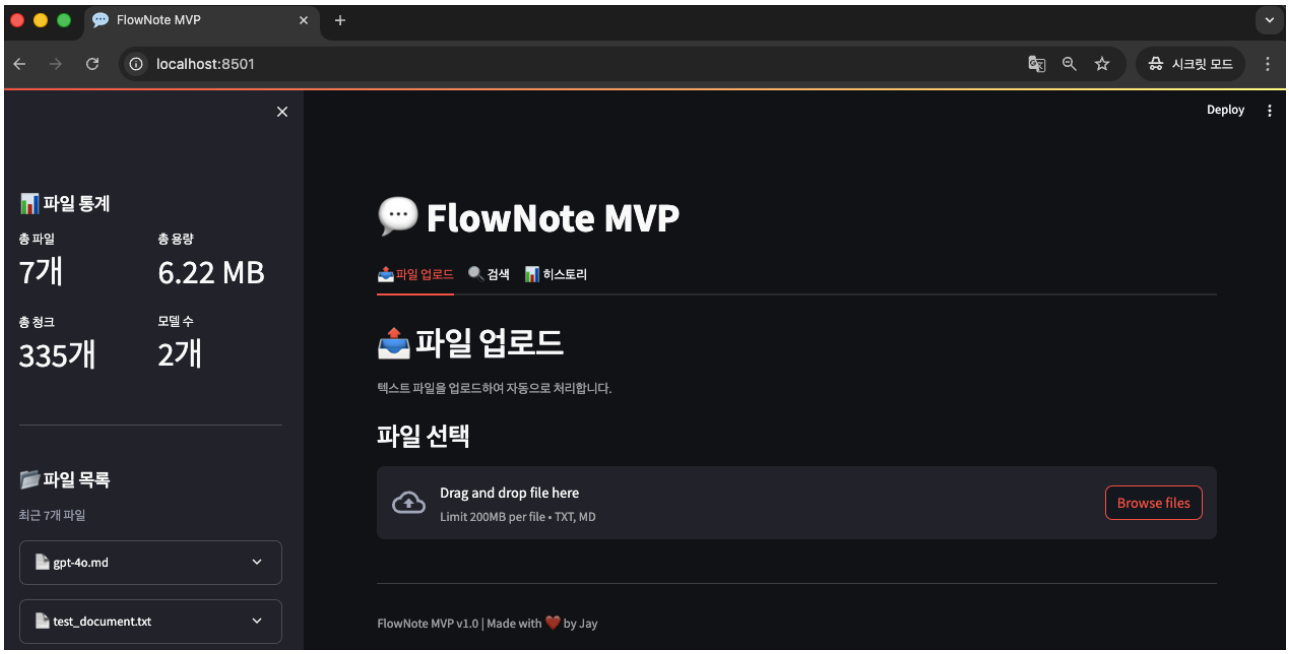
```
streamlit run app.py
```

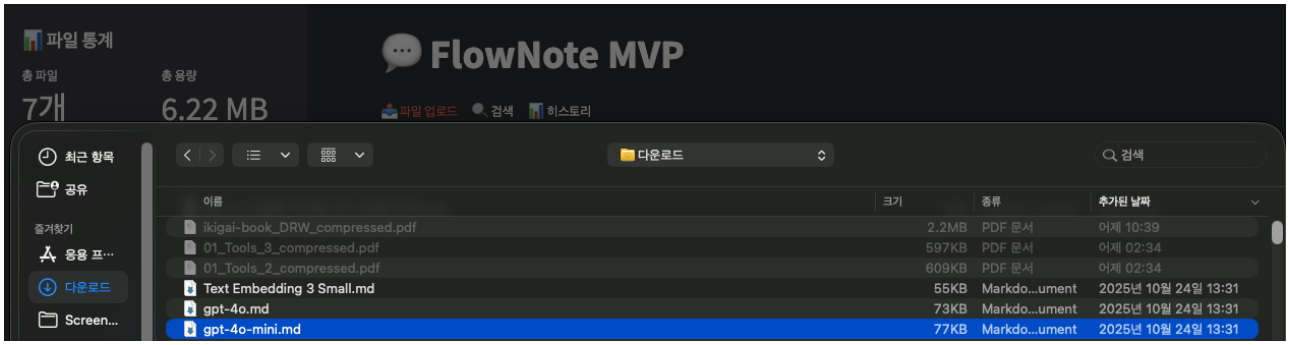
- 브라우저에서 자동으로 `http://localhost:8501` 열림!

📖 사용 방법

Step 1: 파일 업로드

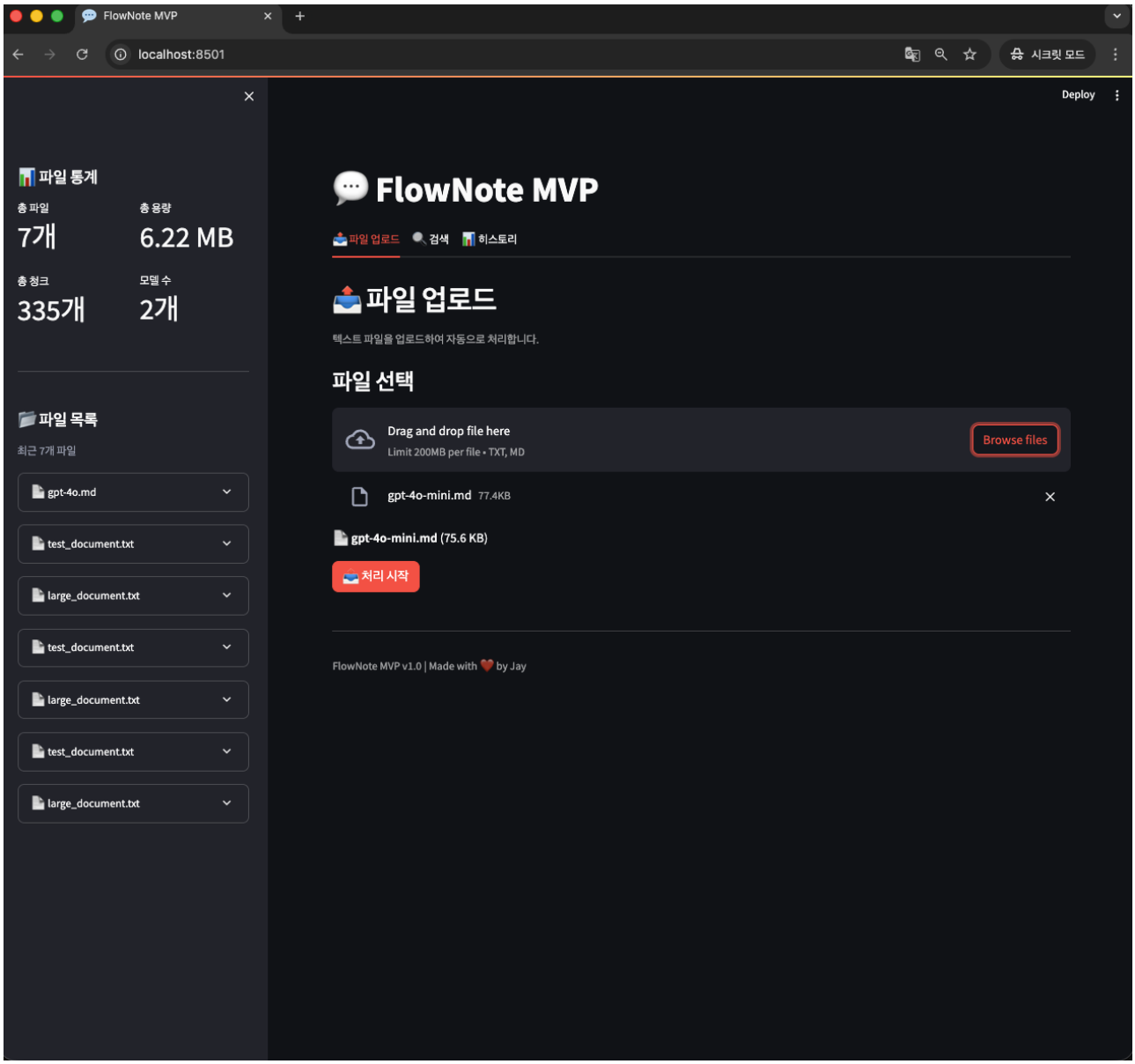
1. **main**에서 📤 **파일 업로드** 클릭

- 

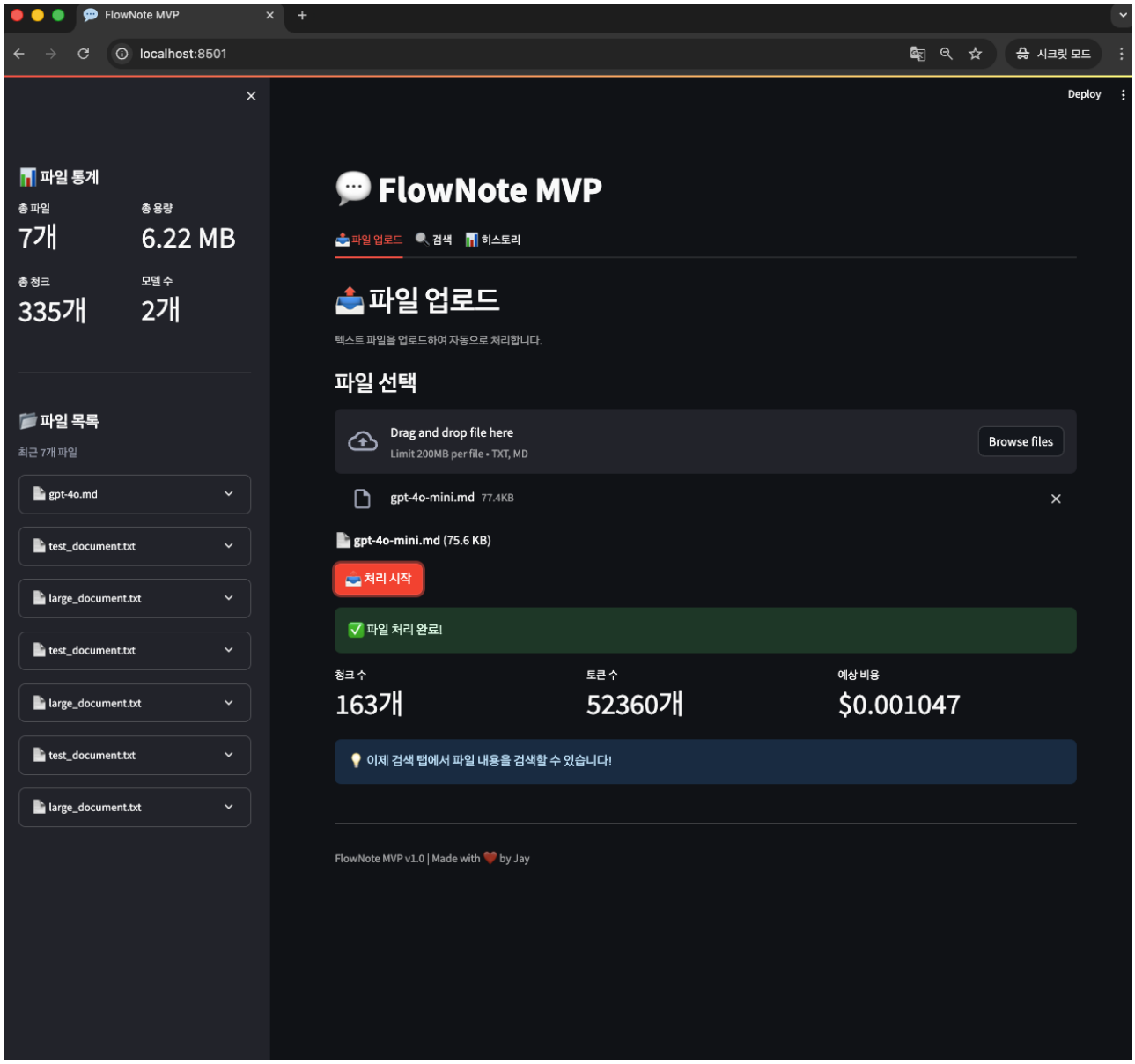
- 

2. AI 대화 파일(`.md`, `.txt`, `.pdf`) 선택 → *미구현, 구현 예정*

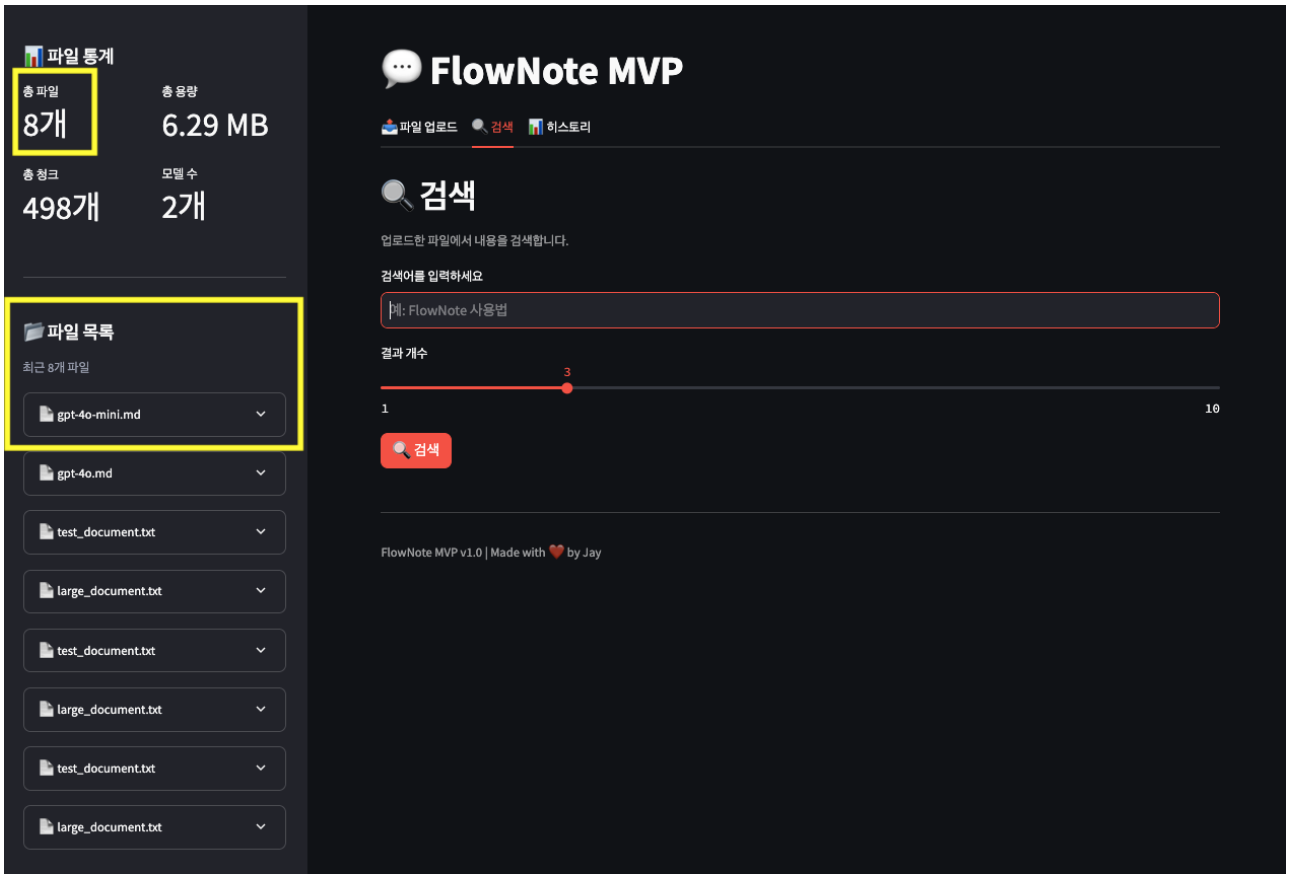
3. **"업로드 & 처리"** 버튼 클릭

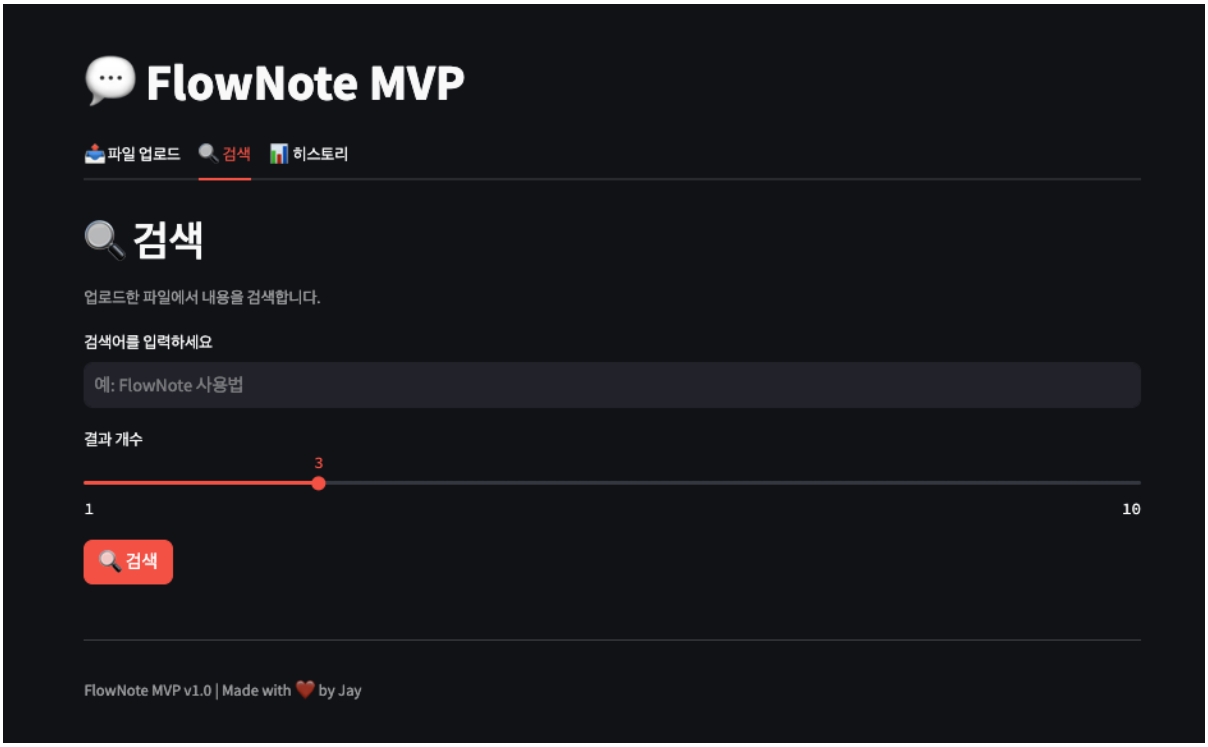


4. 자동으로 청킹 → 임베딩 → FAISS 저장 완료!

- 
FlowNote MVP
localhost:8501
시크릿 모드
Deploy
파일 통계
총 파일
7개
총 용량
6.22 MB
총 청크
335개
모델 수
2개
파일 목록
최근 7개 파일
gpt-4o.md
test_document.txt
large_document.txt
test_document.txt
large_document.txt
test_document.txt
large_document.txt
FlowNote MVP
파일 업로드
검색
히스토리
파일 업로드
텍스트 파일을 업로드하여 자동으로 처리합니다.
파일 선택
Drag and drop file here
Limit 200MB per file • TXT, MD
Browse files
gpt-4o-mini.md 77.4KB
gpt-4o-mini.md (75.6 KB)
처리 시작
파일 처리 완료!
청크 수
163개
토큰 수
52360개
예상 비용
$0.001047
이제 검색 탭에서 파일 내용을 검색할 수 있습니다!
FlowNote MVP v1.0 | Made with by Jay

Step 2: 검색

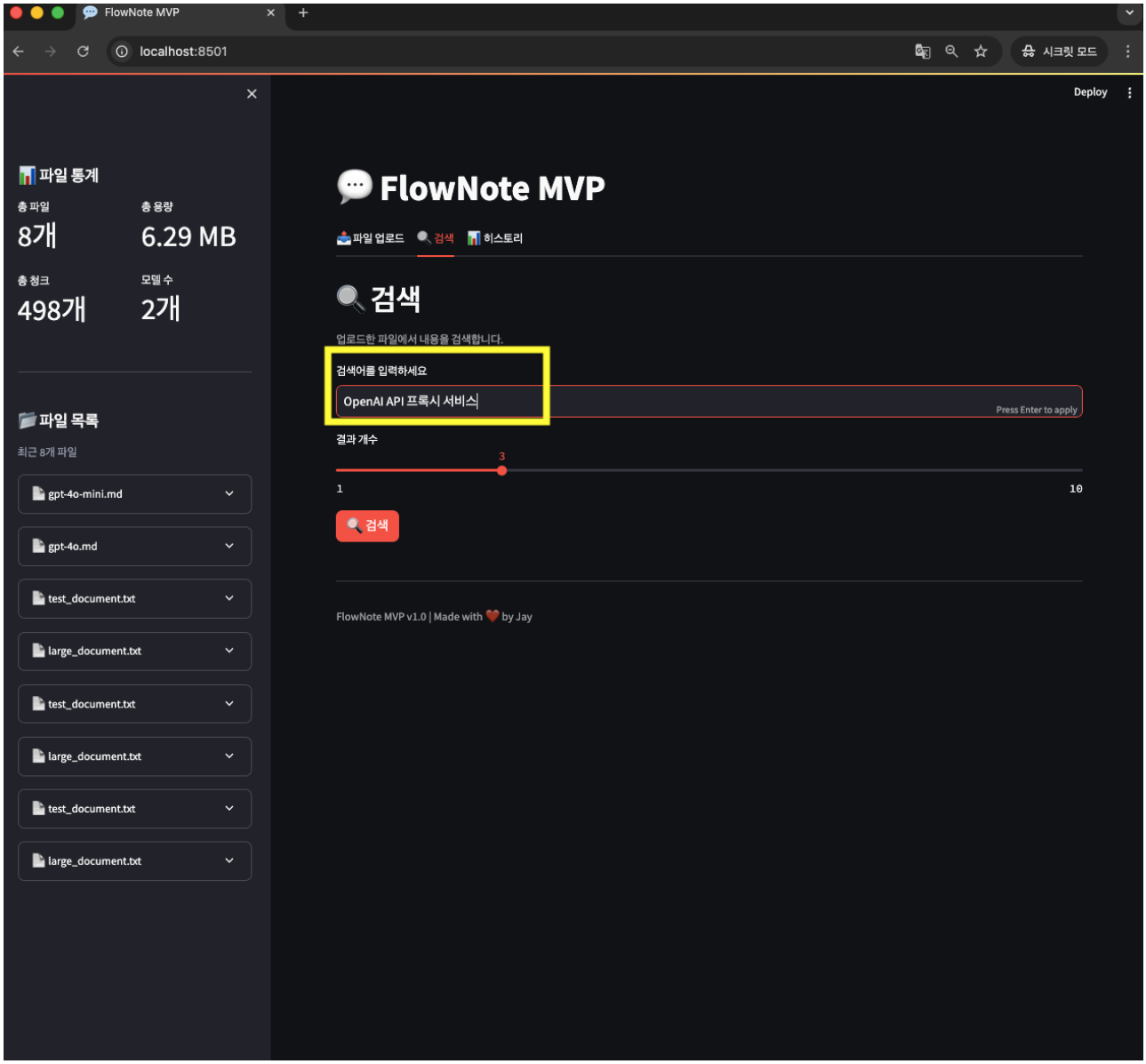


-
    - *최근 업로드한 파일이 좌측 사이드바에 추가된 것을 확인할 수 있음*
- **검색 결과 개수 조절 슬라이더**
    - 

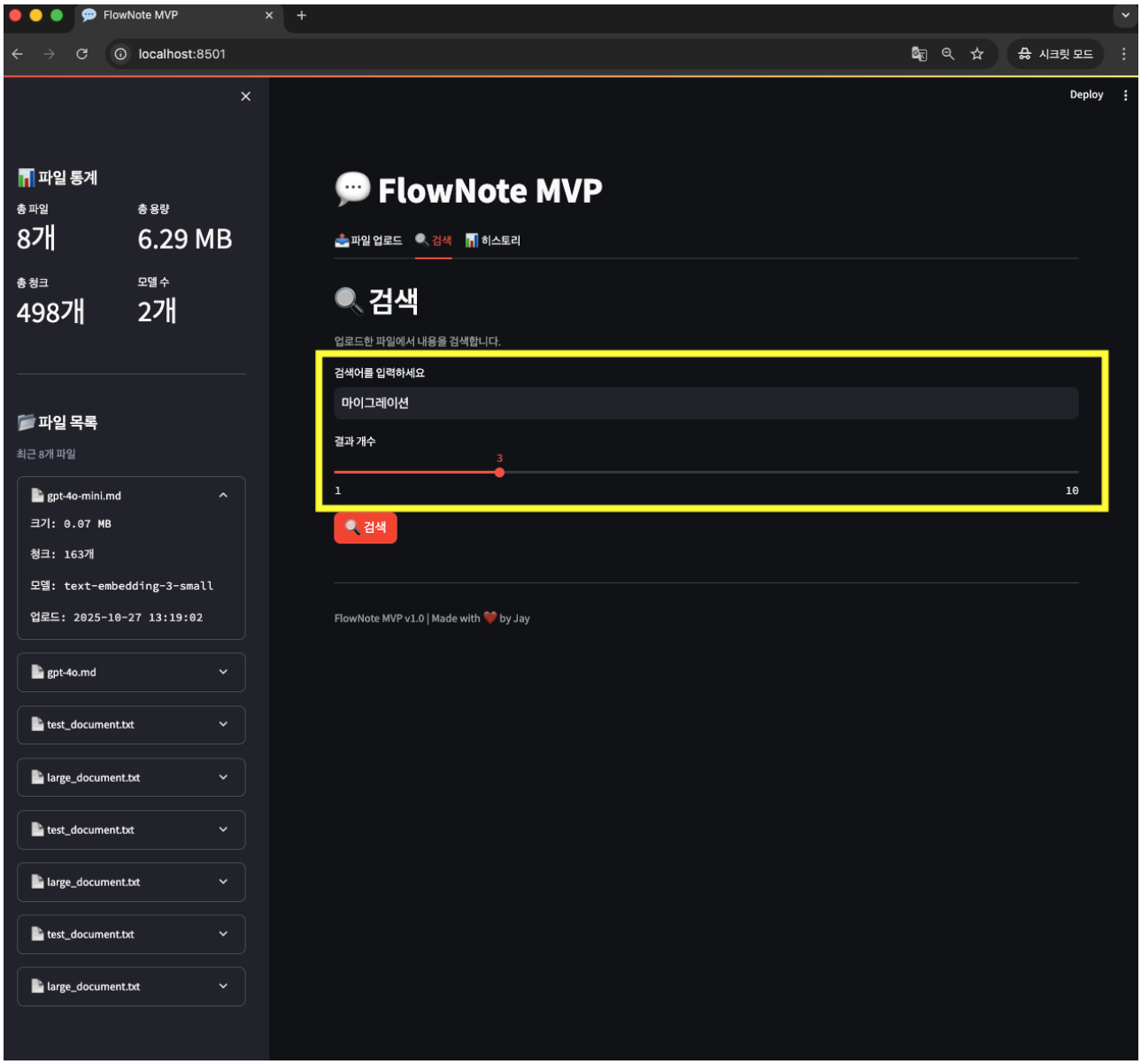
        - 검색 결과 개수: *기본값 = 3*
        - 검색 결과 개수_1
        - 검색 결과 개수 증가 ①: k=5

    - 검색 결과 개수_2
    - 검색 결과 개수 증가 ③: `k=10` / *최대치*
    - 검색 결과 개수_3

1. **메인 화면** 검색창에 질문 입력: *업로드한 파일의 내용에 맞는 내용을 검색하기*
    - 예: *프로젝트 기획 관련 대화 찾아줘*
    - 예: *Python 코드 최적화 방법*

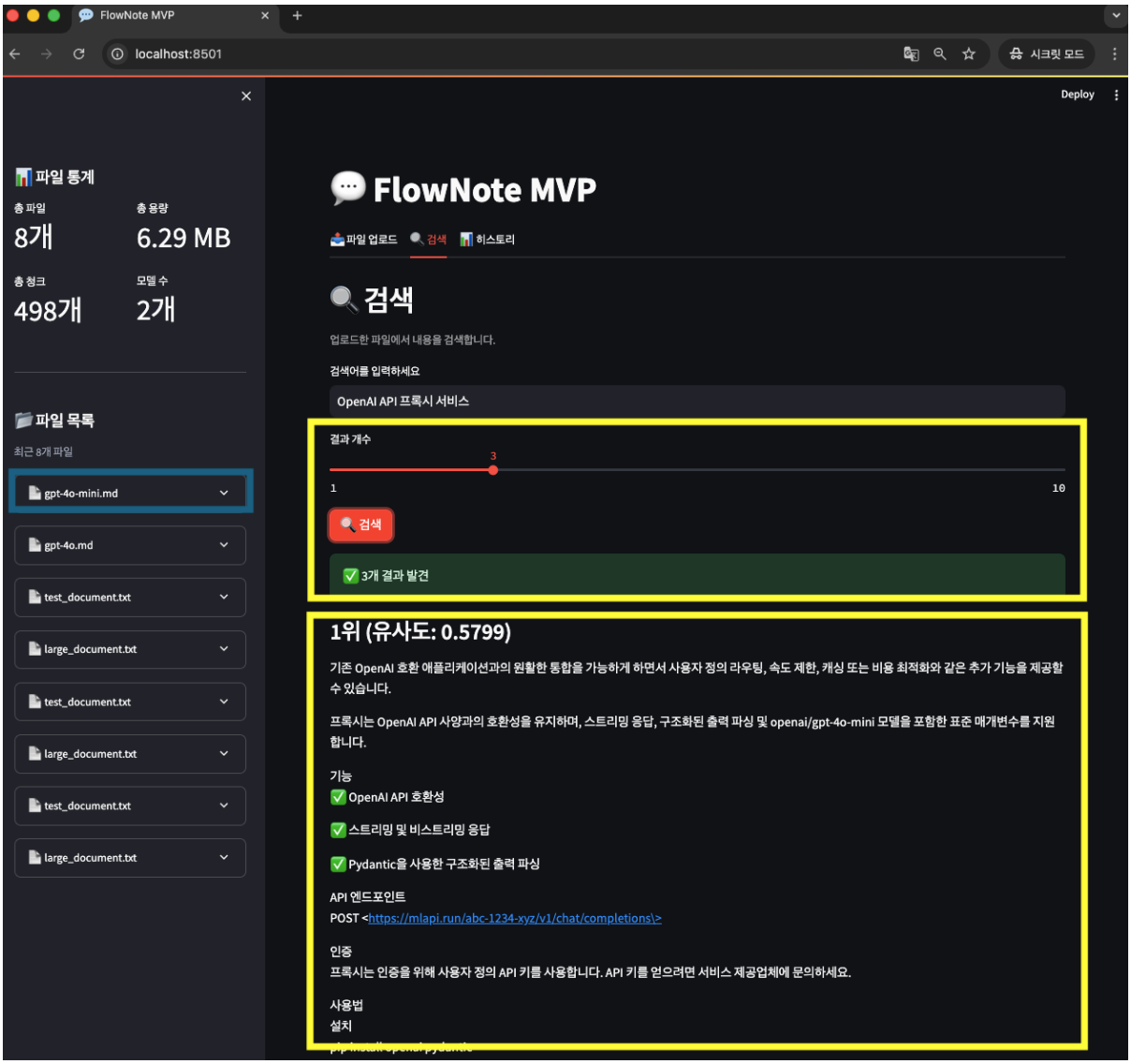
- 검색_①: **`OpenAI API 프록시 서비스`**
- 

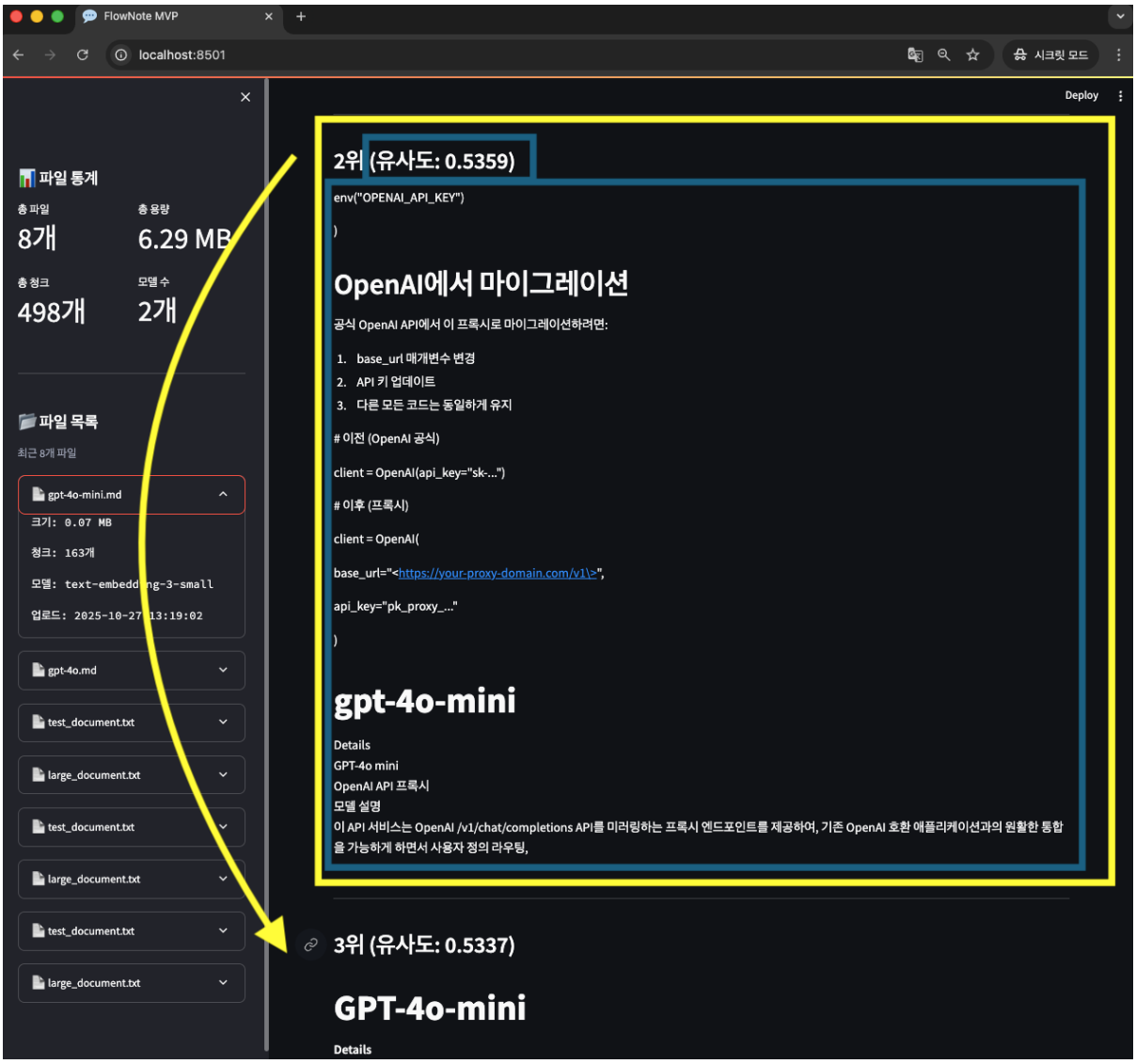
- 검색_②: **`마이그레이션`**

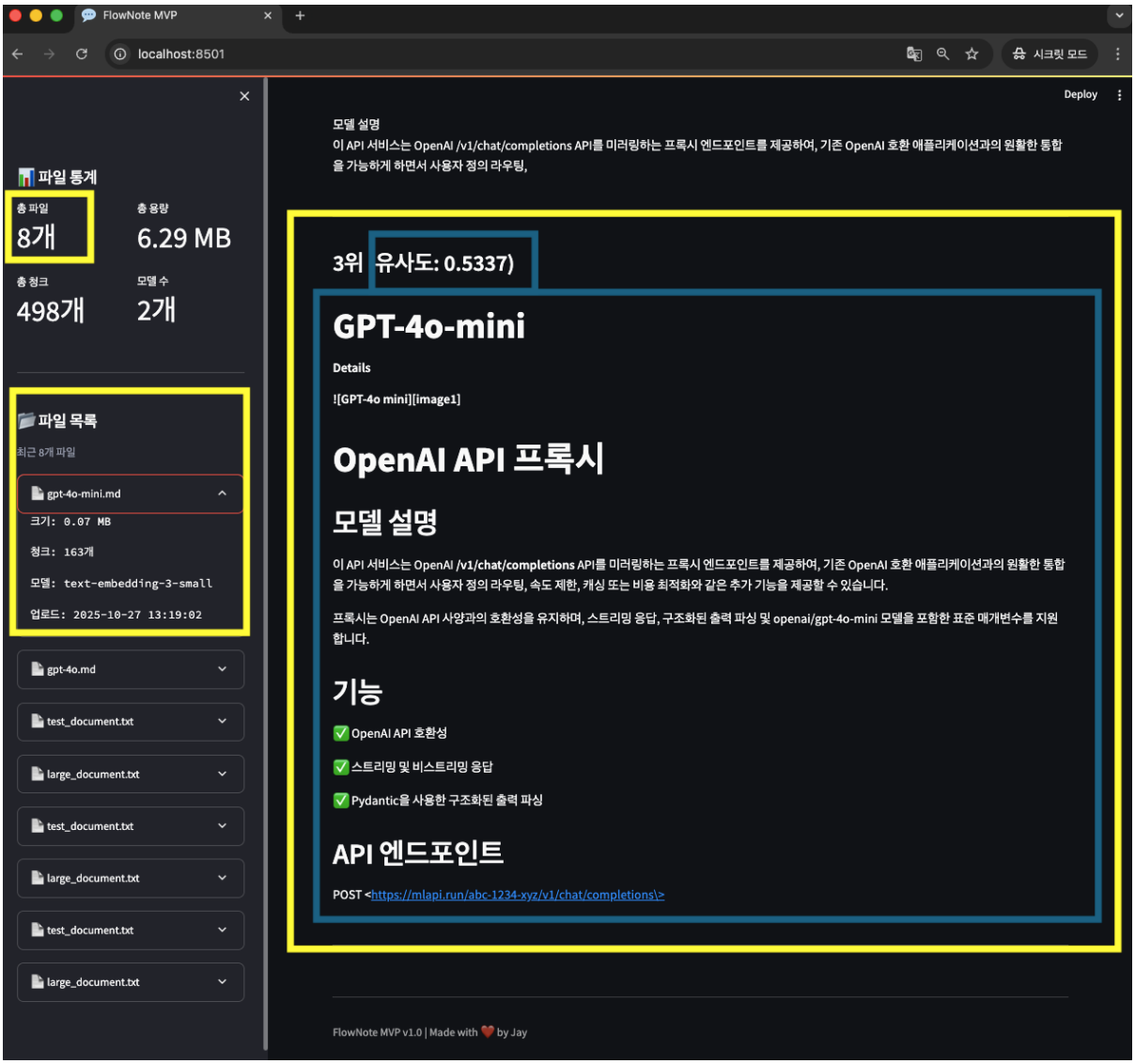
- 

2. 🔍 검색 버튼 클릭
3. 관련도 높은 순으로 결과 표시
4. 각 결과의 유사도 점수와 원본 텍스트 확인

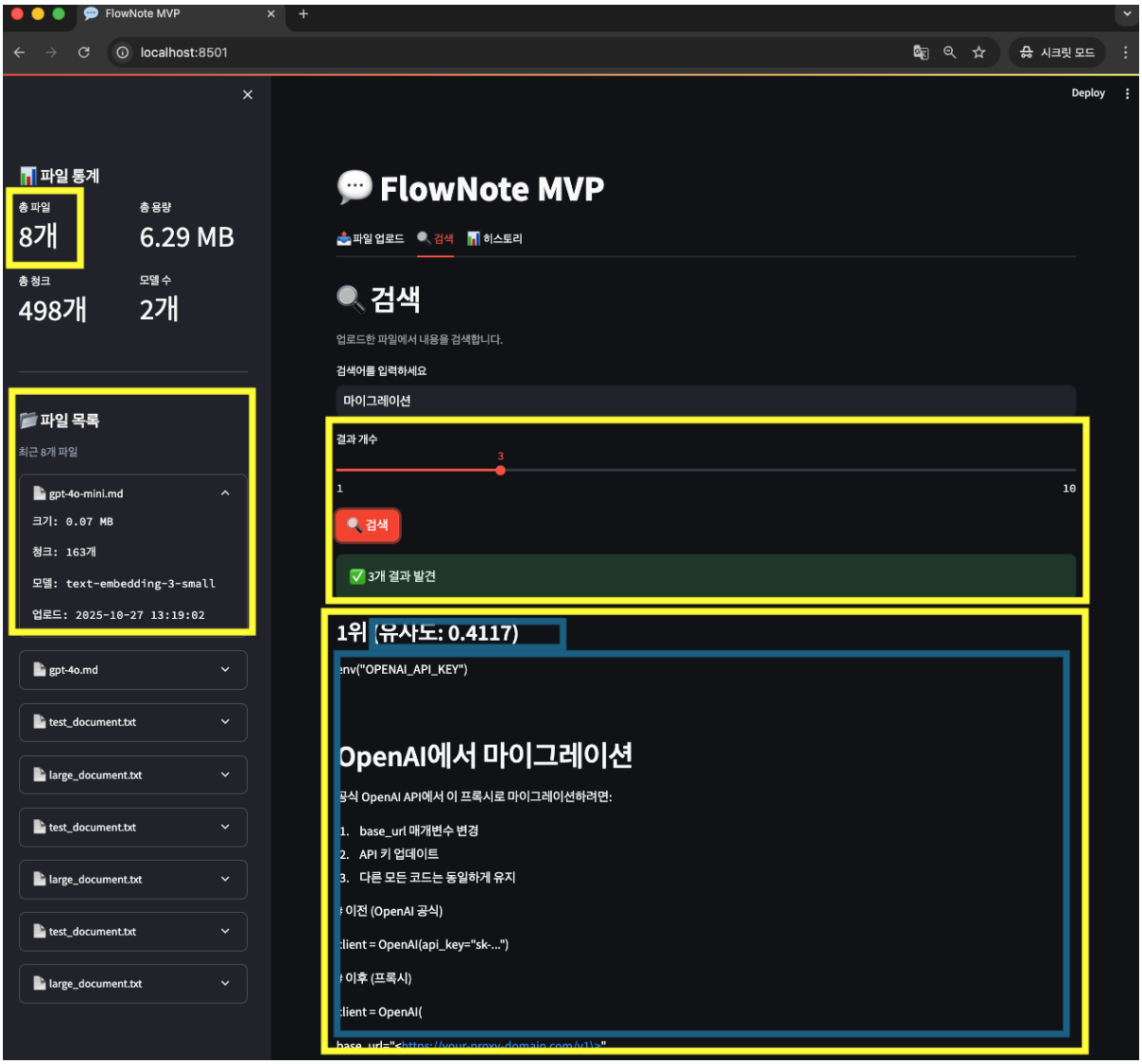
- 검색_①: `OpenAI API 프록시 서비스`
    - 검색 → 관련도 높은 순으로 표기 → 유사도 점수 + 원본 텍스트 확인

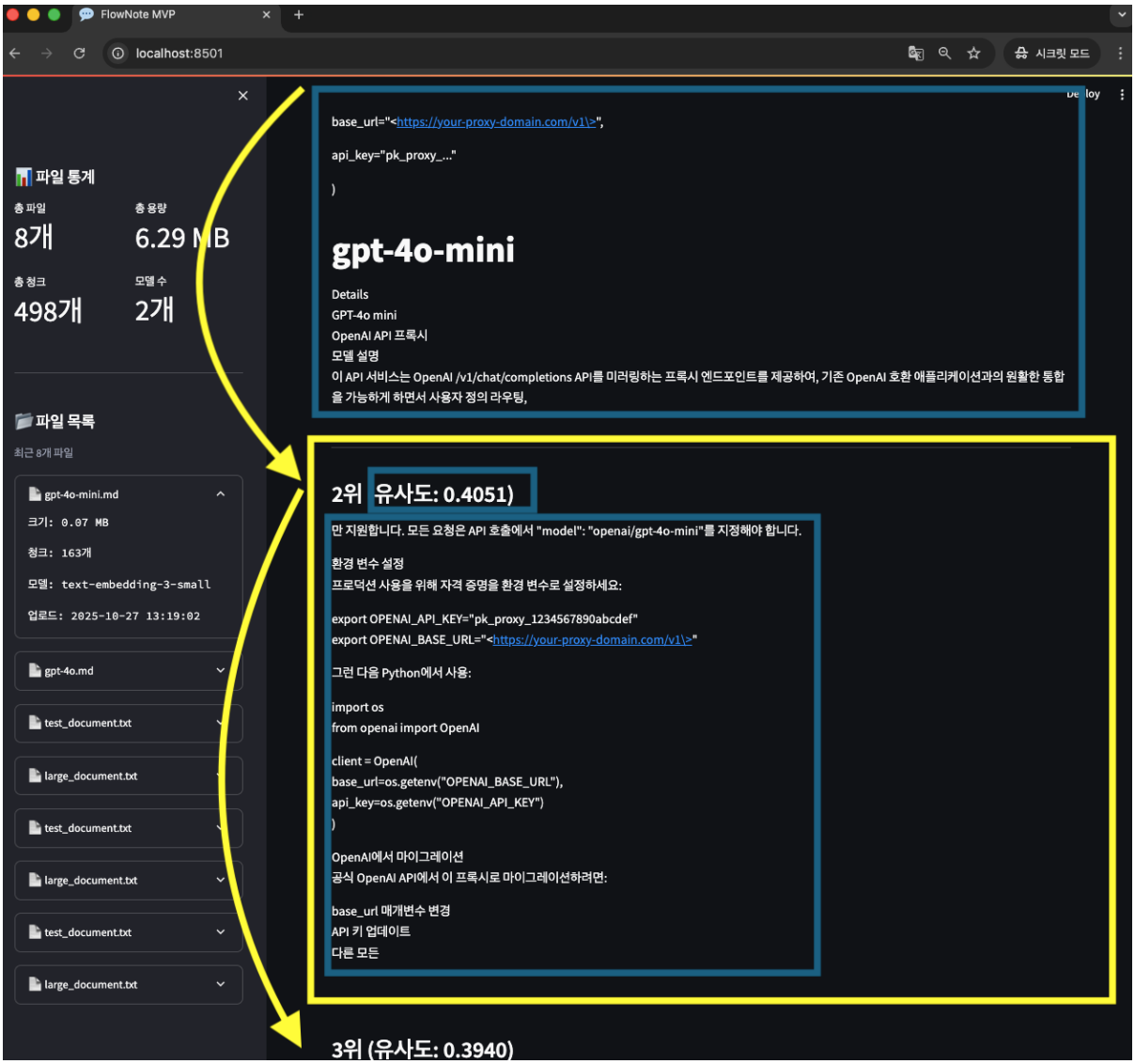

FlowNote MVP
localhost:8501
시크릿 모드
Deploy
파일 통계
총 파일
8개
총 용량
6.29 MB
총 청크
498개
모델 수
2개
파일 목록
최근 8개 파일
gpt-4o-mini.md
gpt-4o.md
test_document.txt
large_document.txt
test_document.txt
large_document.txt
test_document.txt
large_document.txt
FlowNote MVP
파일 업로드
검색
히스토리
검색
업로드한 파일에서 내용을 검색합니다.
검색어를 입력하세요
OpenAI API 프록시 서비스
결과 개수
3
1
10
검색
3개 결과 발견
1위 (유사도: 0.5799)
기존 OpenAI 호환 애플리케이션과의 원활한 통합을 가능하게 하면서 사용자 정의 라우팅, 속도 제한, 캐싱 또는 비용 최적화와 같은 추가 기능을 제공할 수 있습니다.
프록시는 OpenAI API 사양과의 호환성을 유지하며, 스트리밍 응답, 구조화된 출력 파싱 및 openai/gpt-4o-mini 모델을 포함한 표준 매개변수를 지원합니다.
기능
OpenAI API 호환성
스트리밍 및 비스트리밍 응답
Pydantic을 사용한 구조화된 출력 파싱
API 엔드포인트
POST <https://mlapi.run/abc-1234-xyz/v1/chat/completions\>
인증
프록시는 인증을 위해 사용자 정의 API 키를 사용합니다. API 키를 얻으려면 서비스 제공업체에 문의하세요.
사용법
설치

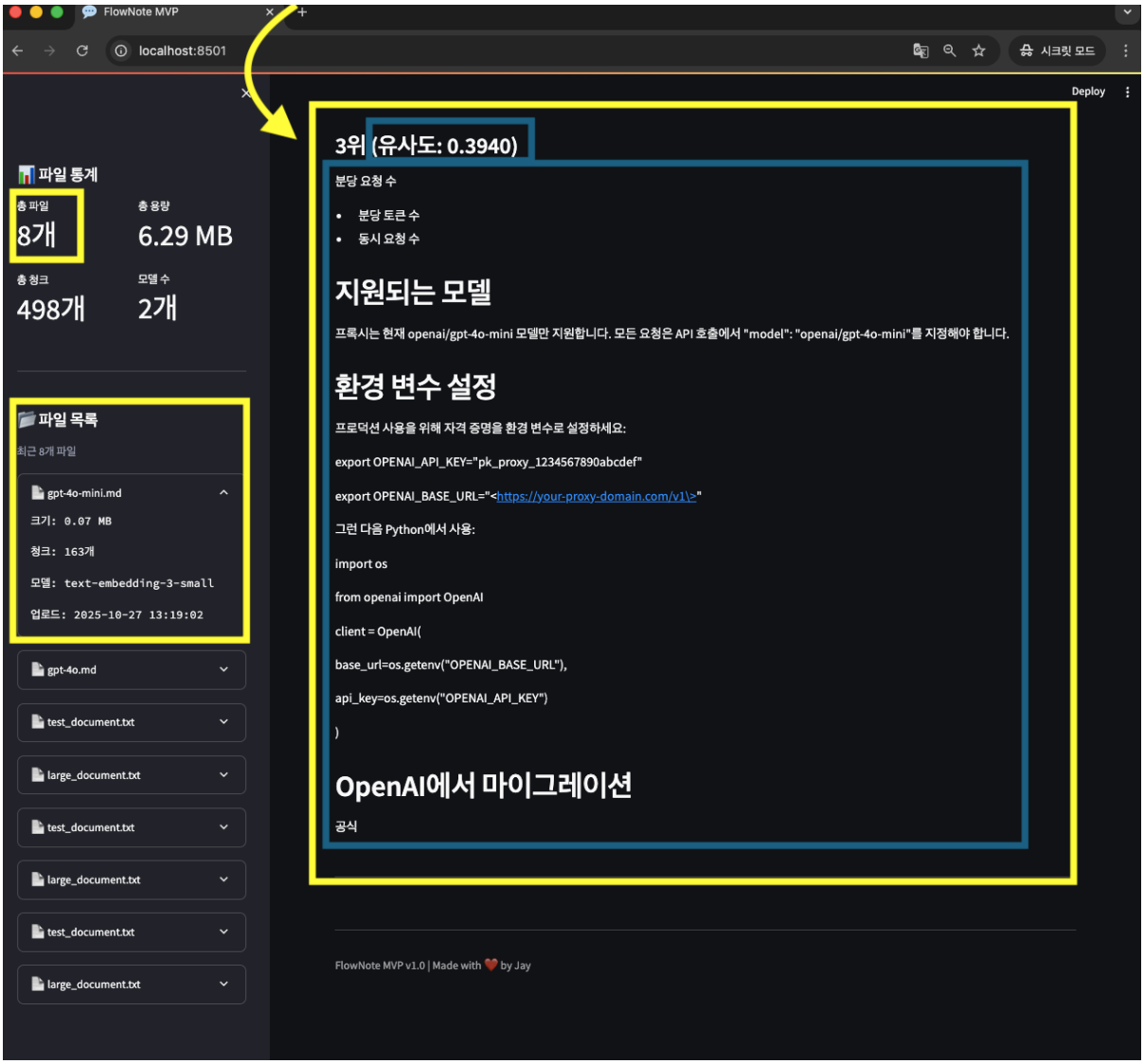

FlowNote MVP
localhost:8501
시크릿 모드
Deploy
파일 통계
총 파일
8개
총 용량
6.29 MB
총 청크
498개
모델 수
2개
파일 목록
최근 8개 파일
gpt-4o-mini.md
크기: 0.07 MB
청크: 163개
모델: text-embedding-3-small
업로드: 2025-10-27 13:19:02
gpt-4o.md
test_document.txt
large_document.txt
test_document.txt
large_document.txt
test_document.txt
large_document.txt
2위 (유사도: 0.5359)
env("OPENAI_API_KEY")
)
OpenAI에서 마이그레이션
공식 OpenAI API에서 이 프록시로 마이그레이션하려면:
1. base_url 매개변수 변경
2. API 키 업데이트
3. 다른 모든 코드는 동일하게 유지
# 이전 (OpenAI 공식)
client = OpenAI(api_key="sk-...")
# 이후 (프록시)
client = OpenAI(
base_url="<https://your-proxy-domain.com/v1\>",
api_key="pk_proxy_..."
)
gpt-4o-mini
Details
GPT-4o mini
OpenAI API 프록시
모델 설명
이 API 서비스는 OpenAI /v1/chat/completions API를 미러링하는 프록시 엔드포인트를 제공하여, 기존 OpenAI 호환 애플리케이션과의 원활한 통합을 가능하게 하면서 사용자 정의 라우팅,
3위 (유사도: 0.5337)
GPT-4o-mini
Details



- 검색_②: 마이그레이션
  - 검색 → 관련도 높은 순으로 표기 → 유사도 점수 + 원본 텍스트 확인


FlowNote MVP
localhost:8501
시크릿 모드
Deploy
파일 통계
총 파일
8개
총 용량
6.29 MB
총 청크
498개
모델 수
2개
파일 목록
최근 8개 파일
gpt-4o-mini.md
크기: 0.07 MB
청크: 163개
모델: text-embedding-3-small
업로드: 2025-10-27 13:19:02
gpt-4o.md
test_document.txt
large_document.txt
test_document.txt
large_document.txt
test_document.txt
large_document.txt
FlowNote MVP
파일 업로드
검색
히스토리
검색
업로드한 파일에서 내용을 검색합니다.
검색어를 입력하세요
마이그레이션
결과 개수
3
1
10
검색
3개 결과 발견
1위 (유사도: 0.4117)
OpenAI에서 마이그레이션
1. base_url 매개변수 변경
2. API 키 업데이트
3. 다른 모든 코드는 동일하게 유지

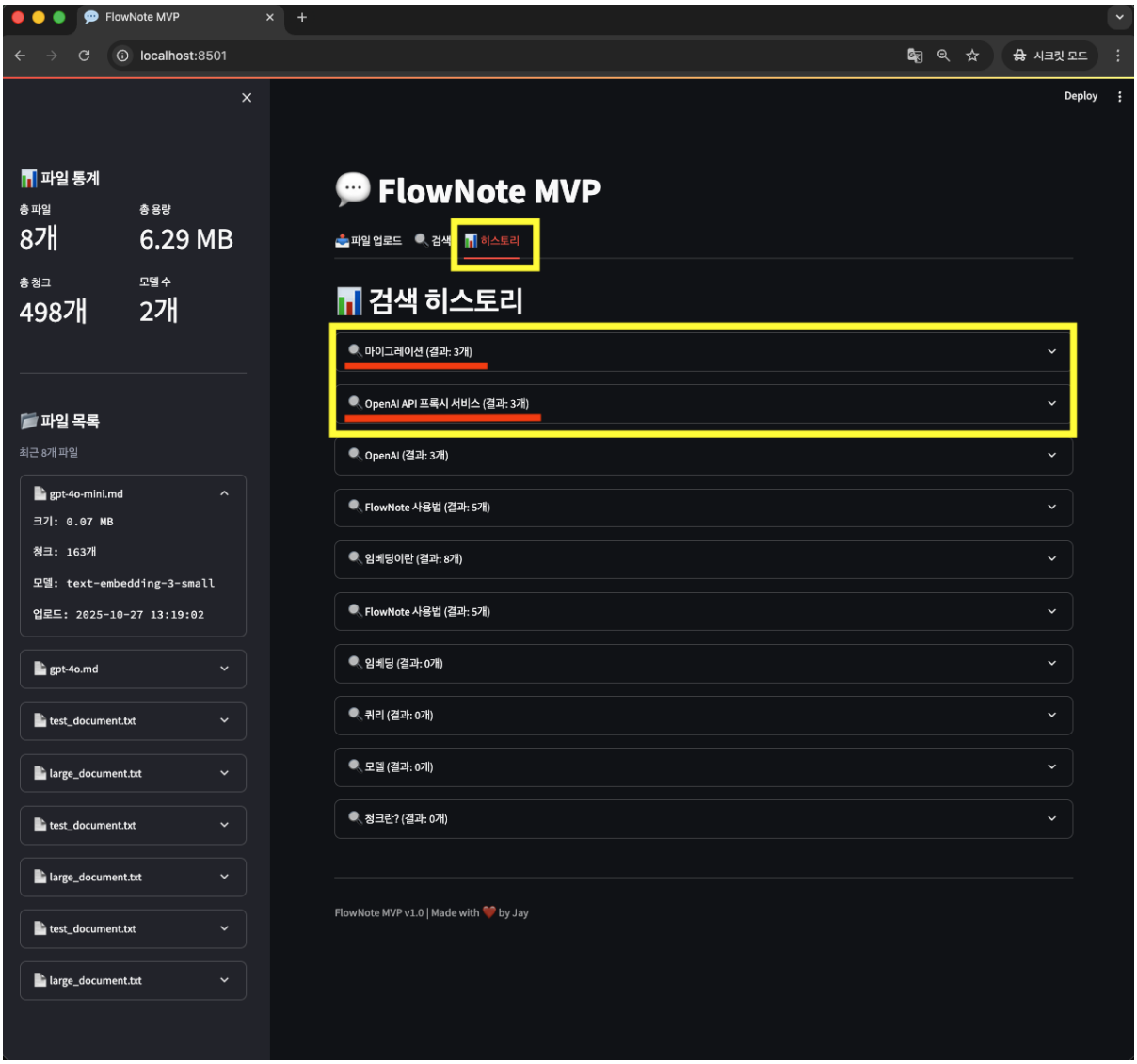

FlowNote MVP
localhost:8501
시크릿 모드
파일 통계
총 파일
8개
총 용량
6.29 MB
총 청크
498개
모델 수
2개
파일 목록
최근 8개 파일
gpt-4o-mini.md
크기: 0.07 MB
청크: 163개
모델: text-embedding-3-small
업로드: 2025-10-27 13:19:02
gpt-4o.md
test_document.txt
large_document.txt
test_document.txt
large_document.txt
test_document.txt
large_document.txt
base_url="<https://your-proxy-domain.com/v1\>",
api_key="pk_proxy_..."
)
gpt-4o-mini
Details
GPT-4o mini
OpenAI API 프록시
모델 설명
이 API 서비스는 OpenAI /v1/chat/completions API를 미러링하는 프록시 엔드포인트를 제공하여, 기존 OpenAI 호환 애플리케이션과의 원활한 통합을 가능하게 하면서 사용자 정의 라우팅,
2위 유사도: 0.4051)
만 지원합니다. 모든 요청은 API 호출에서 "model": "openai/gpt-4o-mini"를 지정해야 합니다.
환경 변수 설정
프로덕션 사용을 위해 자격 증명을 환경 변수로 설정하세요:
export OPENAI_API_KEY="pk_proxy_1234567890abcdef"
export OPENAI_BASE_URL="<https://your-proxy-domain.com/v1\>"
그런 다음 Python에서 사용:
import os
from openai import OpenAI
client = OpenAI(
base_url=os.getenv("OPENAI_BASE_URL"),
api_key=os.getenv("OPENAI_API_KEY")
)
OpenAI에서 마이그레이션
공식 OpenAI API에서 이 프록시로 마이그레이션하려면:
base_url 매개변수 변경
API 키 업데이트
다른 모든
3위 (유사도: 0.3940)



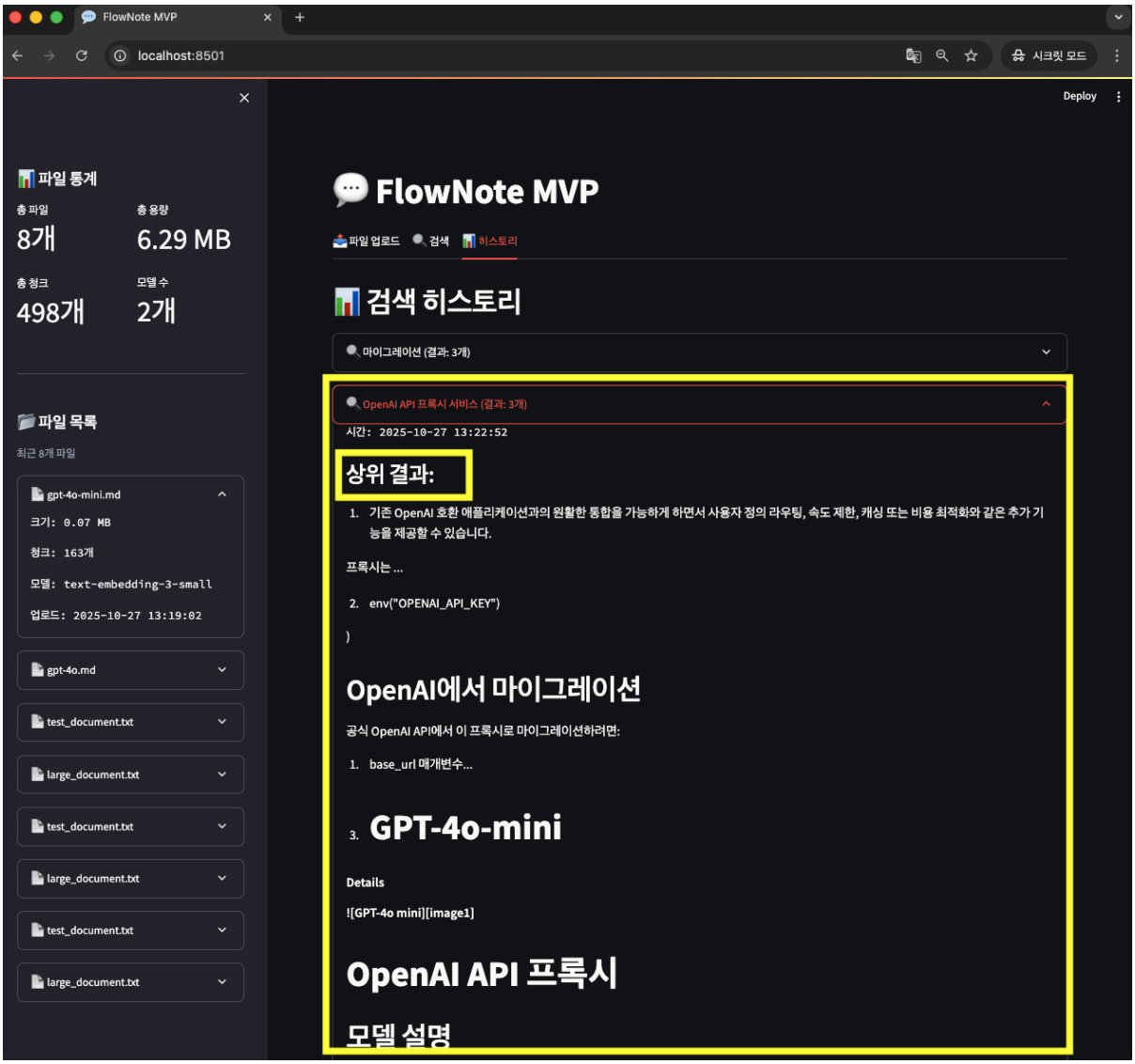
Step 3: 히스토리 활용

- **검색 히스토리**
- **파일 목록**: 업로드한 파일 정보 확인 (최대 10개)

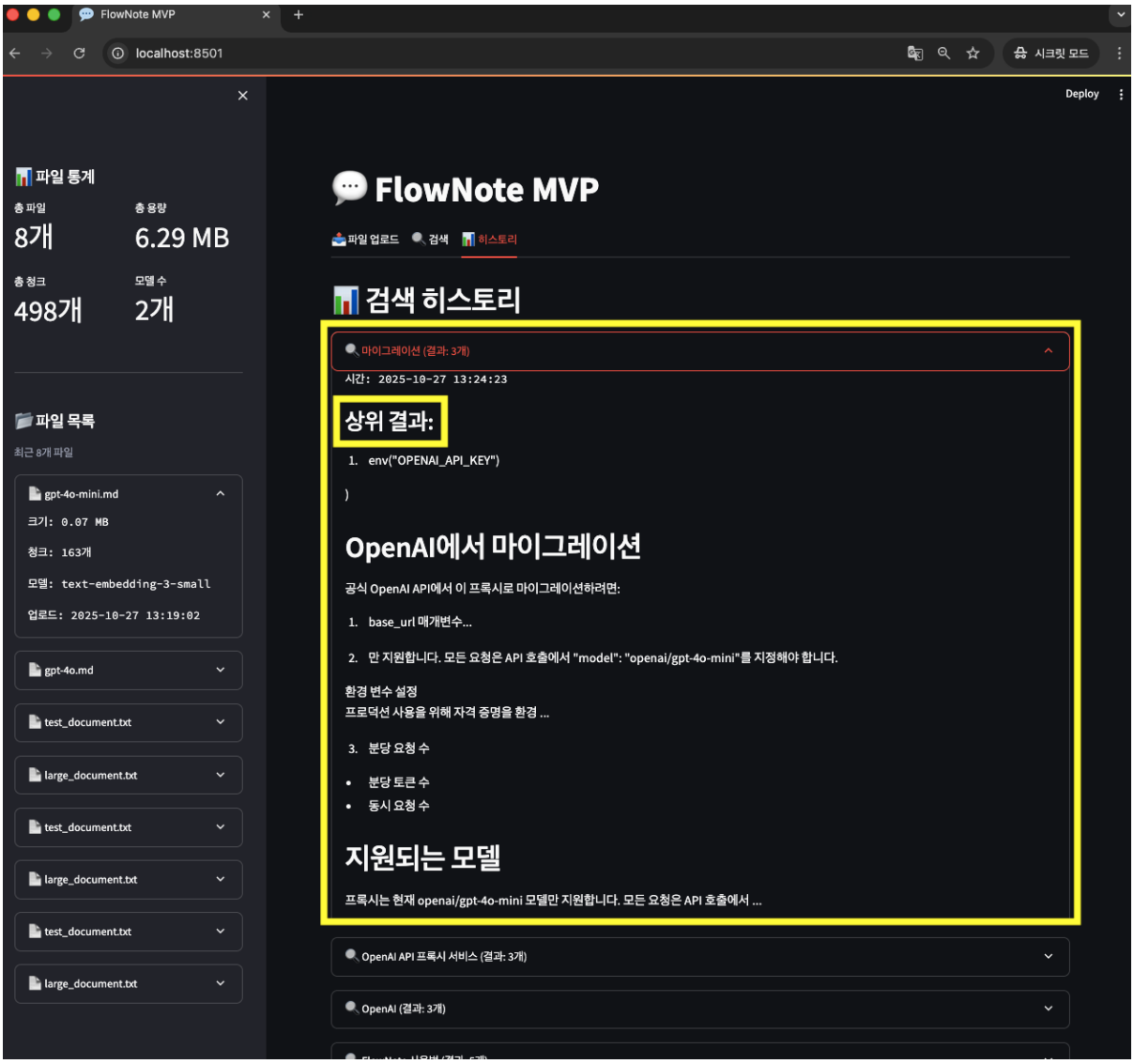
- 검색 히스토리 파일 목록



- 검색_①: `OpenAI API 프록시 서비스`
  - 검색 → 관련도 높은 순으로 표기 → 유사도 점수 + 원본 텍스트 확인



- 검색_②: 마이그레이션
  - 검색 → 관련도 높은 순으로 표기 → 유사도 점수 + 원본 텍스트 확인



📁 프로젝트 구조

```
flownote-mvp/
├── app.py                  # 메인 Streamlit 앱
│
├── requirements.txt        # Python 패키지 의존성
│
├── .env                    # 환경변수 (git 무시)
├── .env.example            # .env 작성 예시 파일
│
├── .python-version         # Python 버전 (3.11.10)
│
├── .gitignore
│
├── README.md
│
├── backend/                # 백엔드 로직
│   ├──__pycache__/         # python 소스 파일의 바이트코드 캐시 저장 폴더
```

```
(.gitignore 처리)
│   │
│   ├── data/
│   │
│   ├── __init__.py
│   ├── config.py           # 설정 관리
│   ├── chunking.py         # 텍스트 분할
│   ├── embedding.py        # 임베딩 생성
│   ├── faiss_search.py     # FAISS 검색
│   ├── metadata.py         # 메타데이터 관리
│   ├── search_history.py   # 검색 히스토리
│   └── utils.py            # 유틸리티 함수
│
├── data/                   # 데이터 저장소
│   │
│   ├── uploads/            # 업로드된 원본 파일 폴더
│   │   ├── 2025-10-24_17-39-17_gpt-4o.md                              #
테스트 업로드 파일 ①
│   │   ├── 2025-10-25_15-05-07_gemini-embedding-troubleshooting.md     #
테스트 업로드 파일 ②
│   │   ├── 2025-10-25_15-05-57_gpt-4o.md                              #
테스트 업로드 파일 ③
│   │   ├── gpt-4o.md                                                  #
테스트 업로드 파일 ④
│   │   └── requirements-mini.md                                       #
테스트 업로드 파일 ⑤
│   │
│   ├── exports/            # 내보낸 파일 (예정)
│   ├── faiss/              # FAISS 인덱스 (에정)
│   ├── metadata/           # 파일 메타데이터 (JSON)
│   └── history/            # 검색 히스토리 (JSON)
│
├── docs/                   # 문서
│   ├── constitution.md     # 프로젝트 헌장
│   │
│   ├── practices/          # backend 파일들 확장 중 과정 기록 폴더
│   │   ├── 2025-10-23-utils-test.md        # 1st 테스트 파일 + Stremlit UI로
확인 정리
│   │   ├── 2025-10-23-utils-test.pdf       # 해당 파일을 pdf 버전
│   │   │
│   │   ├── 2025-10-24-app-py-test.md       # 2nd 테스트 파일 + Stremlit UI로
확인 정리
│   │   ├── 2025-10-24-app-py-test.pdf      # 해당 파일을 pdf 버전
│   │   │
│   │   ├── 2025-10-25-backend-integration-test.md    # 3rd 테스트 파일 +
Stremlit UI로 확인 정리
│   │   └── 2025-10-25-backend-integration-test.pdf   # 해당 파일을 pdf 버전
│   │
│   └── specs/                          # 기능 명세서
│       ├── file-upload.md              # 파일 업로드 관련
│       ├── faiss-search.md             # 검색 관련
│       ├── prompt-templates.md         # 결과 변환 관련 *result_markdown
(string)*
│       ├── markdown-export.md          # 결과 저장 관련 *(data/exports/)*
```

```
│       └── mcp-classification.md       # 업로드된 파일 자동 분류 관련
│
├── assets/                 # 프로젝트에서 사용되는 첨부파일 및 사각자료 등 정적 자원 포함하
는 폴더
│   ├── images/             # 일반 이미지 (스크린샷, 배경 등)
│   │
│   └── figures/            # 문서에 포함되는 분석 결과 이미지
│
└── tests/                  # 테스트 파일 정리 폴더
    ├── __init__.py
    ├── test_all_models.py                  # gpt-4o, gpt-4o-mini, text-
embedding-3-small 호출 확인 테스트
    ├── test_chunking_embedding.py          # chunking_embedding 확인 테스트
    ├── test_faiss.py                       # faiss 확인 테스트
    └── test_new_models.py                  # 모델 추가 및 호출 확인 테스트
                                            # text-embedding-3-large,
claude-3.5-haiku, claude-4-sonnet
```

🛠️ 기술 스택

Core Technologies

분류	기술	설명
Frontend	`Streamlit`	빠르고 직관적인 웹 UI
Embedding	OpenAI API	`text-embedding-3-small` 모델
Vector DB	FAISS	빠른 로컬 벡터 검색
Text Processing	LangChain	텍스트 분할 & 처리
File Parsing	PyMuPDF, python-docx	PDF, Word 파일 파싱
Language	`Python 3.11.10`	안정적인 최신 버전

Key Libraries

```
streamlit>=1.28.0
openai>=1.3.0
faiss-cpu>=1.7.4
langchain>=0.0.340
python-dotenv>=1.0.0
pymupdf>=1.23.0
python-docx>=1.0.0
```

🎯 주요 기능 상세

1. 파일 처리 파이프라인

```
파일 업로드
    ↓
텍스트 추출 (PyMuPDF, python-docx)
    ↓
청킹 (RecursiveCharacterTextSplitter)
├─ chunk_size: 1000
└─ chunk_overlap: 200
    ↓
임베딩 생성 (OpenAI text-embedding-3-small)
    ↓
FAISS 인덱스 추가
    ↓
메타데이터 저장
```

2. 검색 프로세스

```
사용자 질문 입력
    ↓
질문 임베딩 생성
    ↓
FAISS 유사도 검색 (Top K=3)
    ↓
유사도 점수 계산 (0-100%)
    ↓
결과 표시 & 히스토리 저장
```

3. 메타데이터 관리

각 파일별로 다음 정보를 자동 기록:

- **파일명**: 원본 파일 이름
- **크기**: 파일 크기 (KB/MB)
- **청크 수**: 분할된 텍스트 조각 개수
- **임베딩 모델**: 사용한 임베딩 모델명
- **업로드 시간**: ISO 8601 형식 타임스탬프
- **파일 ID**: 고유 식별자

📊 성능 & 제약

⚡ 성능

- **검색 속도**: 평균 0.3초 이하 (1000개 청크 기준)
- **임베딩 속도**: 파일당 2-5초 (크기에 따라)
- **동시 검색**: FAISS 인메모리 처리로 빠른 응답

⚠ 현재 제약사항 (v1.0)

- **파일 크기 제한**: 최대 200MB (Streamlit 기본값)
- **FAISS 영속화**: 앱 재시작 시 인덱스 초기화 (v1.5에서 개선 예정)
- **동시 사용자**: 로컬 단일 사용자만 지원
- **검색 결과**: 최대 5개까지 표시

🔮 로드맵

✅ v1.0 MVP (현재)

- [x] 파일 업로드 & 처리
- [x] FAISS 벡터 검색
- [x] 검색 히스토리
- [x] 메타데이터 관리
- [x] 기본 UI/UX

🚧 v1.5 (11/5-11/16) - 프로젝트 버전

- [ ] **LangChain 통합**: ConversationalRetrievalChain
- [ ] **대화 기억**: Memory 기능 추가
- [ ] **답변 생성**: 검색 결과 기반 AI 답변
- [ ] **출처 인용**: 답변에 출처 자동 표시
- [ ] **FAISS 영속화**: 인덱스 저장/로드
- [ ] **에러 핸들링**: 안정성 강화
- [ ] **파일별/날짜별 필터링**: 검색 결과 필터링 기능

🔮 v2.0 (12월 이후) - 확장 버전

- [ ] Notion 연동
- [ ] 옵시디언 Export
- [ ] 자동 요약 & 인사이트
- [ ] 카테고리 자동 분류
- [ ] 다국어 지원
- [ ] 멀티 사용자 지원

🤝 기여하기

FlowNote는 오픈소스 프로젝트입니다! 기여를 환영합니다.

기여 방법

1. 이 리포지토리를 **Fork**
2. 새 브랜치 생성 (`git checkout -b feature/amazing-feature`)
3. 변경사항 커밋 (`git commit -m 'feat: Add amazing feature'`)
4. 브랜치에 푸시 (`git push origin feature/amazing-feature`)
5. **Pull Request** 생성

이슈 제보

- 버그 발견: Issue 생성
- 기능 제안: Discussion

📚 문서 & 리소스

- **프로젝트 헌장**: docs/constitution.md
- **기능 명세서**: docs/specs/
- **이슈 트래커**: GitHub Issues
- **프로젝트 보드**: GitHub Projects

❓ FAQ

Q1. API 비용이 많이 드나요?

- OpenAI `text-embedding-3-small` 모델은 매우 저렴합니다 ($0.00002/1K tokens). 일반적인 사용(하루 10개 파일 업로드)으로는 월 $1 미만입니다.

Q2. 데이터는 어디에 저장되나요?

- 모든 데이터는 **로컬**에 저장됩니다 (`data/` 폴더). 아직 외부 서버로 전송되지 않습니다. (OpenAI API 제외)

Q3. 다른 임베딩 모델을 사용할 수 있나요?

- 현재는 OpenAI만 지원하지만, `backend/embedding.py`를 수정하면 HuggingFace 등 다른 모델 사용 가능합니다.

Q4. 검색이 느려요.

- FAISS 인덱스가 커질수록 약간 느려질 수 있습니다. v1.5에서 최적화 예정입니다.

Q5. 앱을 재시작하면 데이터가 사라져요.

- 현재 v1.0에서는 FAISS 인덱스가 인메모리로만 저장됩니다. 앱을 재시작하면 다시 업로드해야 합니다.

해결 방법:

- 메타데이터는 `data/metadata/`에 JSON으로 저장되어 유지됩니다
- 원본 파일은 `data/uploads/`에 보관됩니다

- v1.5에서 FAISS 영속화로 재업로드 불필요하게 개선 예정입니다

📄 라이선스

이 프로젝트는 **MIT License** 하에 배포됩니다.

자세한 내용은 LICENSE 파일을 참조하세요.

👤 개발자

Jay

- GitHub: @jjaayy2222

🙏 감사의 말

이 프로젝트는 다음 라이브러리와 도구들 덕분에 가능했습니다:

- Streamlit - 빠른 웹 앱 개발
- LangChain - LLM 애플리케이션 프레임워크
- FAISS - 빠른 벡터 검색
- OpenAI - 강력한 임베딩 모델
- Perplexity AI - 개발 과정의 조력자
- Claude - `Claude-4.5-sonnet` 모델

FlowNote - AI 대화를 흘려보내지 않고, 기록하고 활용하세요 💙

Made with ❤️ by Jay