

FlowNote MVP - Backend 통합 테스트 결과

날짜: 2025년 10월 25일

테스트 환경: Python 3.11 + Streamlit

목적: Backend 모듈 통합 및 UI 연동 검증

1. 🎯 테스트 개요

Backend 핵심 모듈들의 통합 및 Streamlit UI와의 연동을 검증함

테스트 범위

- ✅ Backend 모듈 단독 실행
- ✅ 모듈 간 의존성 검증
- ✅ Streamlit UI 렌더링
- ⚠️ 파일 업로드 → 검색 연동 (미완성)

2. 🛠 Backend 모듈 테스트

1) Config 모듈

```
python -m backend.config
```

=====

설정 확인

=====

📁 프로젝트 경로:

- ROOT: /Users/jay/ICT-projects/flownote-mvp
- DATA: /Users/jay/ICT-projects/flownote-mvp/data

🔑 API 키 확인:

- GPT40: ✅ 설정됨
- GPT40_MINI: ✅ 설정됨
- EMBEDDING: ✅ 설정됨
- EMBEDDING_LARGE: ✅ 설정됨

🤖 모델:

- GPT40: openai/gpt-4o
- GPT40_MINI: openai/gpt-4o-mini
- EMBEDDING: text-embedding-3-small
- EMBEDDING_LARGE: openai/text-embedding-3-large

✓

임베딩 클라이언트 생성 성공!

=====

- 결과: ✓ 정상
- 검증 항목:
 - API 키 로드
 - BASE_URL 설정
 - 클라이언트 생성

2) Utils 모듈

```
python -m backend.utils
```

=====

유틸리티 함수 테스트

=====

텍스트: FlowNote는 AI 대화 관리 도구입니다.
토큰 수: 13
예상 비용: \$0.000000
파일 크기 예시: 1.5 KB

=====

- 결과: ✓ 정상
- 검증 항목:
 - 토큰 계산 (`count_tokens`)
 - 비용 추정 (`estimate_cost`)
 - 파일 크기 포맷 (`format_file_size`)

3) Chunking 모듈

```
python -m backend.chunking
```

=====

청킹 테스트

=====

✅ 생성된 청크 수: 4

- 첫 청크: FlowNote는 AI 대화 관리 도구입니다. FlowNote는 AI 대화 관리 도구입니다...

✅ 메타데이터 포함 청크: 4개

- 첫 청크 정보: {'text': 'FlowNote는 AI 대화 관리 도구입니다. FlowNote는 AI 대화 관리 도구입니다. FlowNote는 AI 대화 관리 도구입니다. FlowNote는 AI 대화 관리 도구입니다.', 'metadata': {'source': 'test'}, 'chunk_index': 0, 'total_chunks': 4}

=====

• 결과: ✅ 정상

• 검증 항목:

- 텍스트 분할 (chunk_text)
- 메타데이터 포함 (chunk_with_metadata)
- TextChunker 클래스

4) Embedding 모듈

```
python -m backend.embedding
```

=====

임베딩 테스트

=====

📊 임베딩 생성 중... (3개 청크)

✅ 임베딩 완료!

- 청크 수: 3
- 토큰 수: 48
- 예상 비용: \$0.000001
- 벡터 차원: 1536

=====

• 결과: ✅ 정상

• 검증 항목:

- 임베딩 생성 (generate_embeddings)
- 토큰 계산 통합
- 비용 추정 통합
- EmbeddingGenerator 클래스

5) FAISS Search 모듈

```
python -m backend.faiss_search
```

```
=====
FAISS 검색 테스트
=====
```

```
✅ 문서 추가 완료!
```

```
- 총 문서 수: 3
```

```
- 인덱스 크기: 3
```

```
검색 결과:
```

```
1위:
```

```
- 유사도: 0.5385
```

```
- 텍스트: 대화 내용을 검색하고 분석할 수 있습니다.
```

```
2위:
```

```
- 유사도: 0.4233
```

```
- 텍스트: 마크다운으로 대화를 내보낼 수 있습니다.
```

```
=====
```

- 결과: ✅ 정상
- 검증 항목:
 - FAISS 인덱스 생성
 - 문서 추가 (`add_documents`)
 - 유사도 검색 (`search`)
 - FAISSRetriever 클래스

6) Metadata 모듈

```
python -m backend.metadata
```

```
=====
파일 메타데이터 테스트
=====
```

```
1. 파일 추가 테스트
```

```
=====
✅ 파일 추가 완료: file_20251025_151445_84fb1fd3
```

✅ 파일 추가 완료: file_20251025_151445_52a6b101

2. 파일 조회 테스트

📄 첫 번째 파일:

- 파일명: test_document.txt
- 크기: 0.05 MB
- 청크 수: 10
- 모델: text-embedding-3-small

📄 두 번째 파일:

- 파일명: large_document.txt
- 크기: 2.0 MB
- 청크 수: 50
- 모델: text-embedding-3-large

3. 전체 파일 목록

📁 등록된 파일: 6개

- file_20251025_131227_d9977552: test_document.txt
- file_20251025_131227_2e480777: large_document.txt
- file_20251025_145527_16a6f607: test_document.txt
- file_20251025_145527_edb1679e: large_document.txt
- file_20251025_151445_84fb1fd3: test_document.txt
- file_20251025_151445_52a6b101: large_document.txt

4. 통계 테스트

📊 통계:

- 총 파일: 6개
- 총 청크: 180개
- 총 크기: 6.15 MB
- 사용 모델: ['text-embedding-3-small', 'text-embedding-3-large']

=====

테스트 완료!

=====

- 결과: ✅ 정상
- 참고: 테스트 실행할 때마다 데이터 누적됨 (정상 동작)
- 검증 항목:
 - 파일 추가 (add_file)
 - 파일 조회 (get_file)
 - 전체 목록 (list_files)
 - 통계 계산 (get_stats)
 - FileMetadata 클래스

7) Search History 모듈

```
python -m backend.search_history
```

```
=====
검색 히스토리 테스트
=====
```

1. 검색 기록 추가 테스트

- ✅ 검색 추가 완료: search_20251025_151552_3c85657e
- ✅ 검색 추가 완료: search_20251025_151552_bd489c20
- ✅ 검색 추가 완료: search_20251025_151552_c5bbe03f

2. 최근 검색 조회 테스트

📖 최근 검색 5개:

1. FlowNote 사용법

- 결과: 5개
- 시간: 2025-10-25 15:15:52

2. 임베딩이란

- 결과: 8개
- 시간: 2025-10-25 15:15:52

3. FlowNote 사용법

- 결과: 5개
- 시간: 2025-10-25 15:15:52

4. 임베딩

- 결과: 0개
- 시간: 2025-10-25 15:09:10


5. 쿼리

- 결과: 0개
- 시간: 2025-10-25 15:08:59

3. 통계 테스트

📊 통계:

- 총 검색: 13회
- 평균 결과: 4.2개
- 자주 검색: FlowNote 사용법

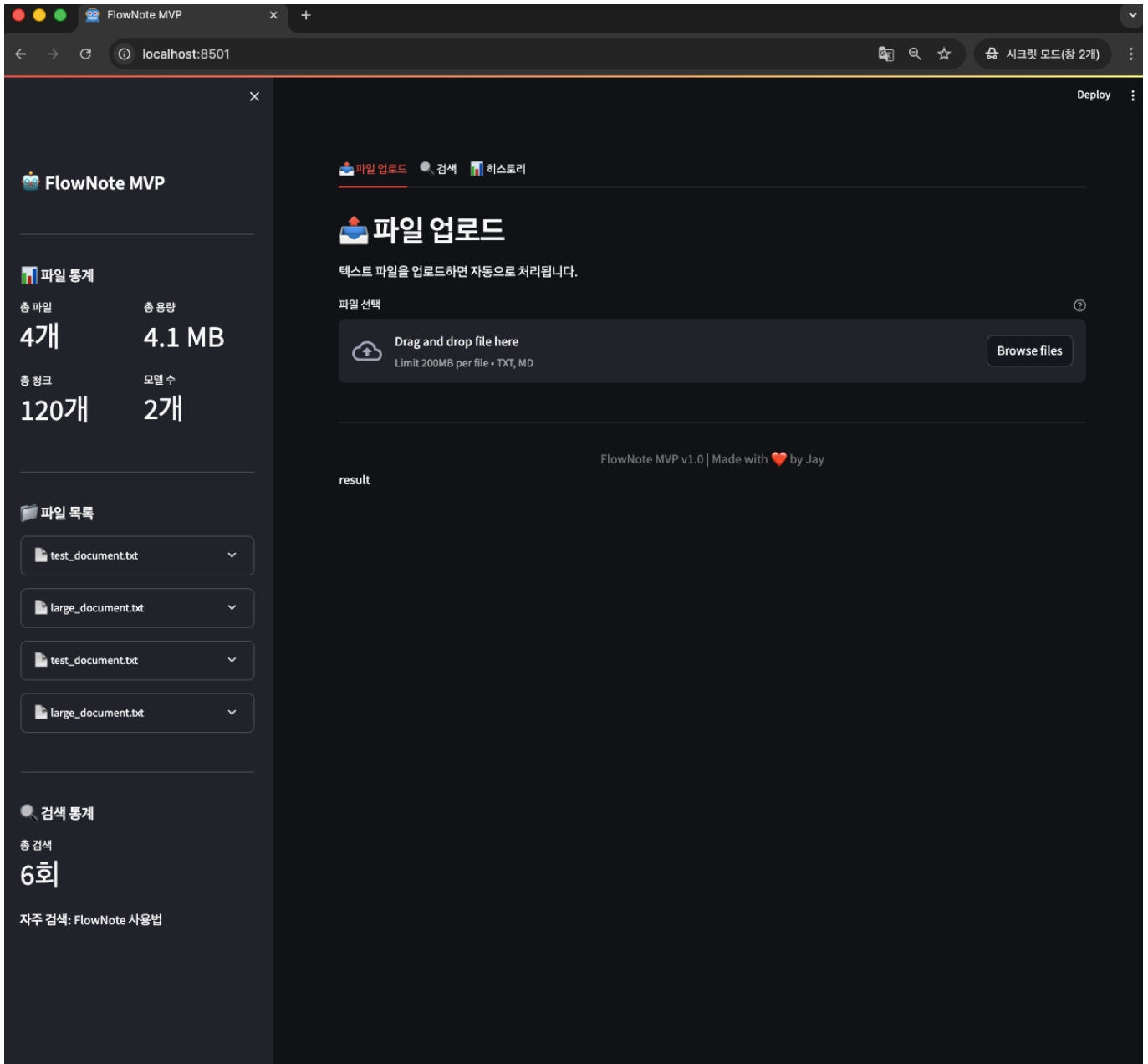
- 결과:  정상
 - 검증 항목:
 - 검색 기록 추가 (`add_search`)
 - 최근 검색 조회 (`get_recent`)
 - 통계 계산 (`get_stats`)
 - SearchHistory 클래스
-

3. Streamlit UI 테스트

1) 실행 방법

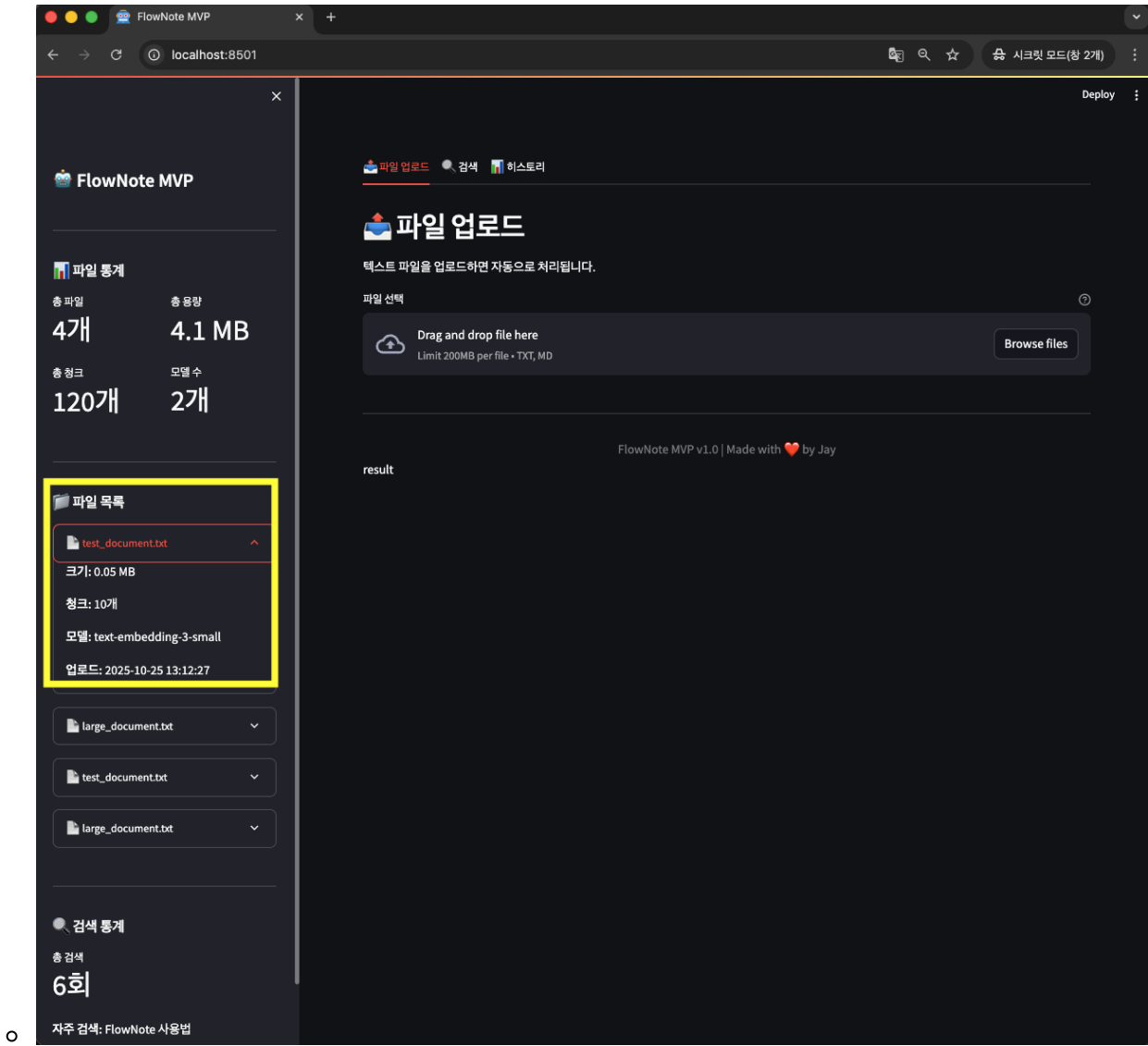
```
streamlit run app.py
```

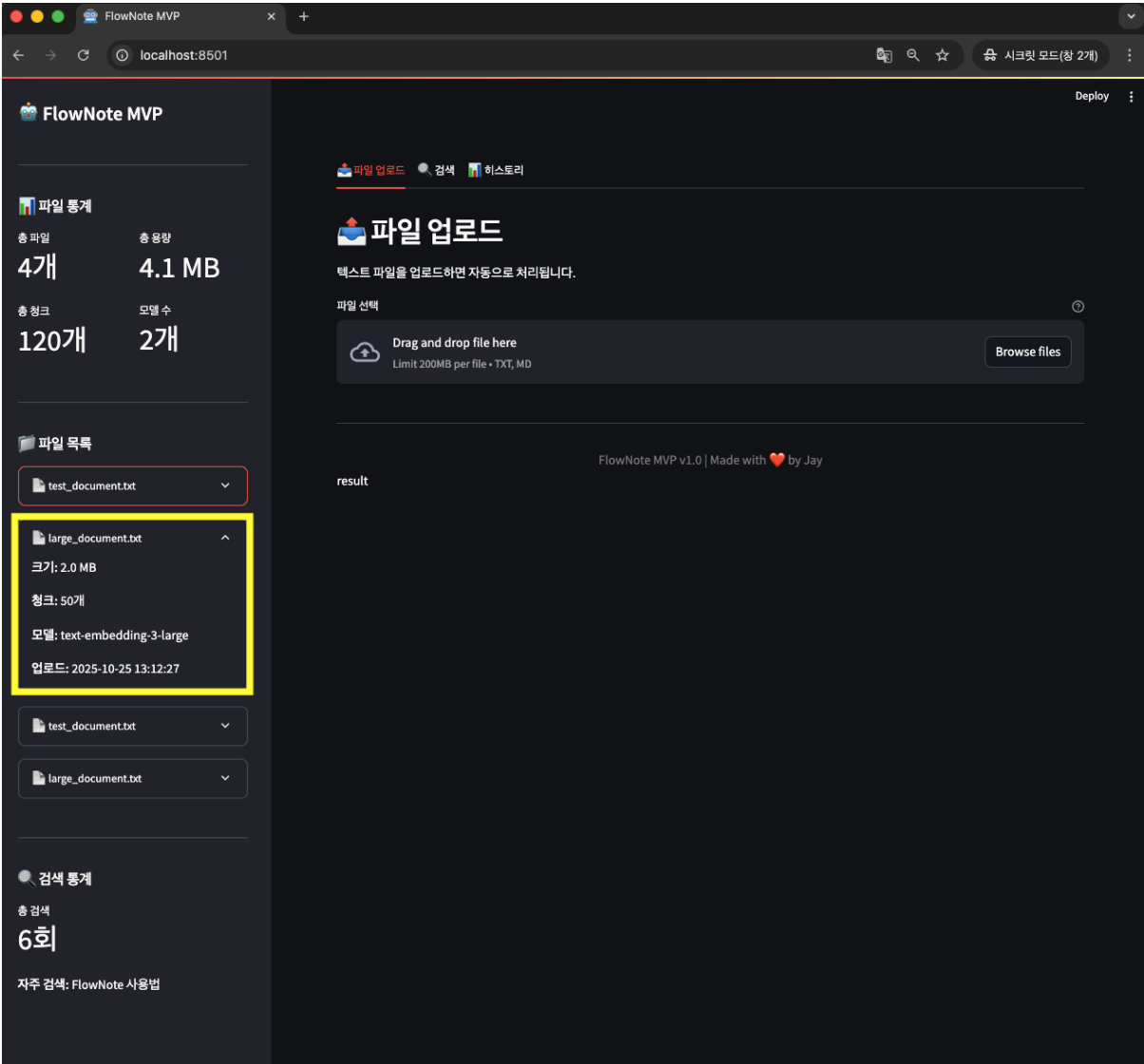
2) 테스트 결과

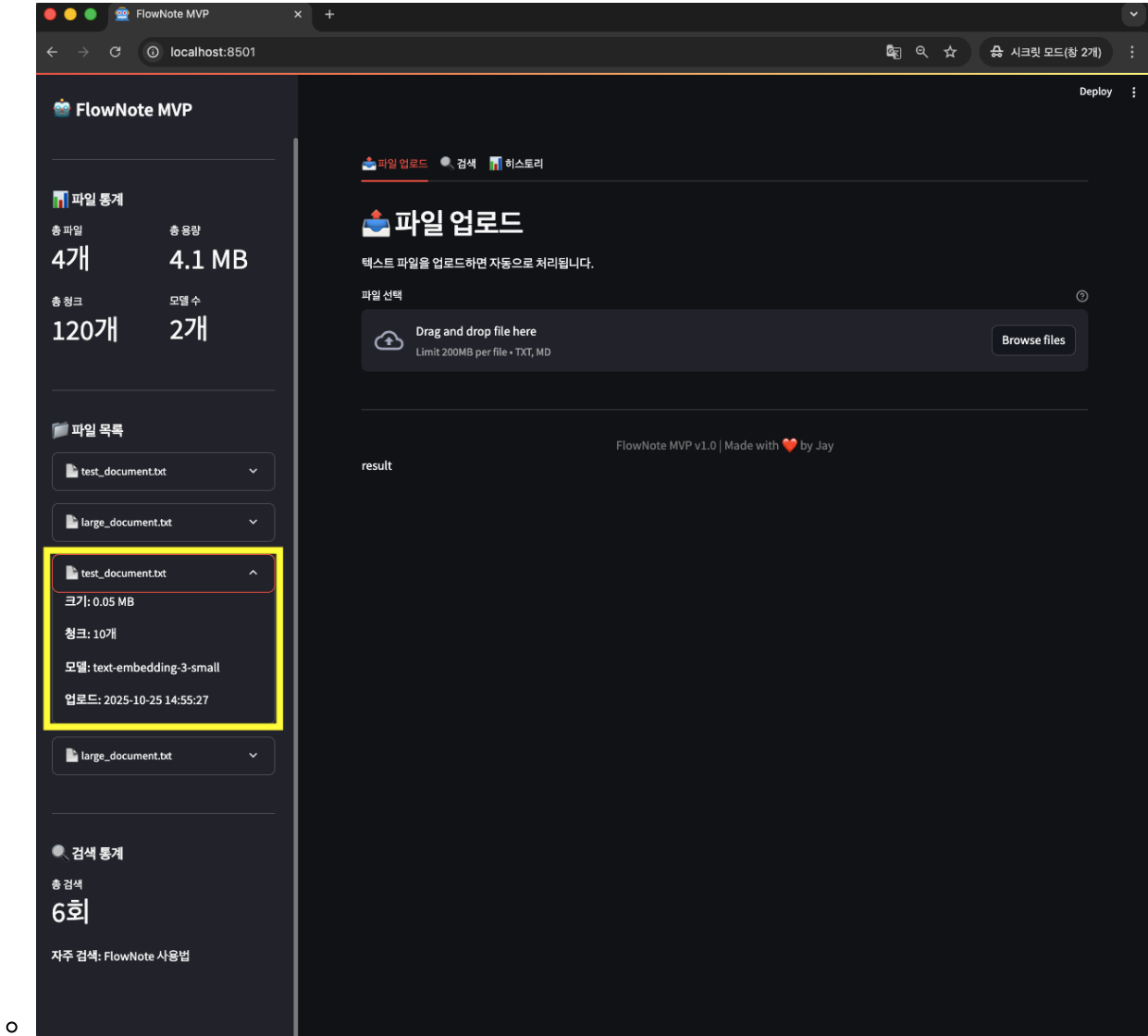


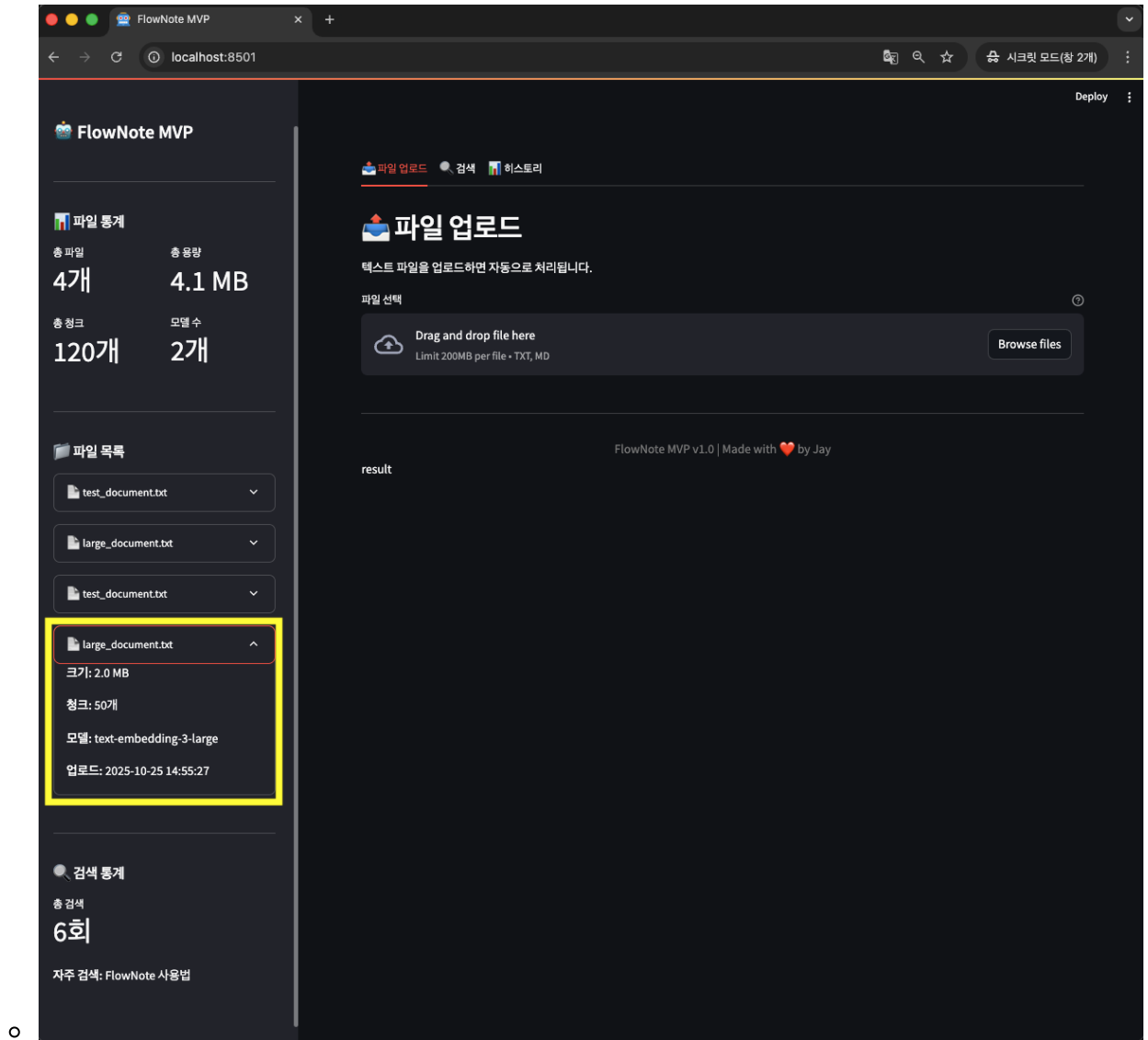
① 파일 통계 (사이드바)

- ✅ 총 파일: 4개





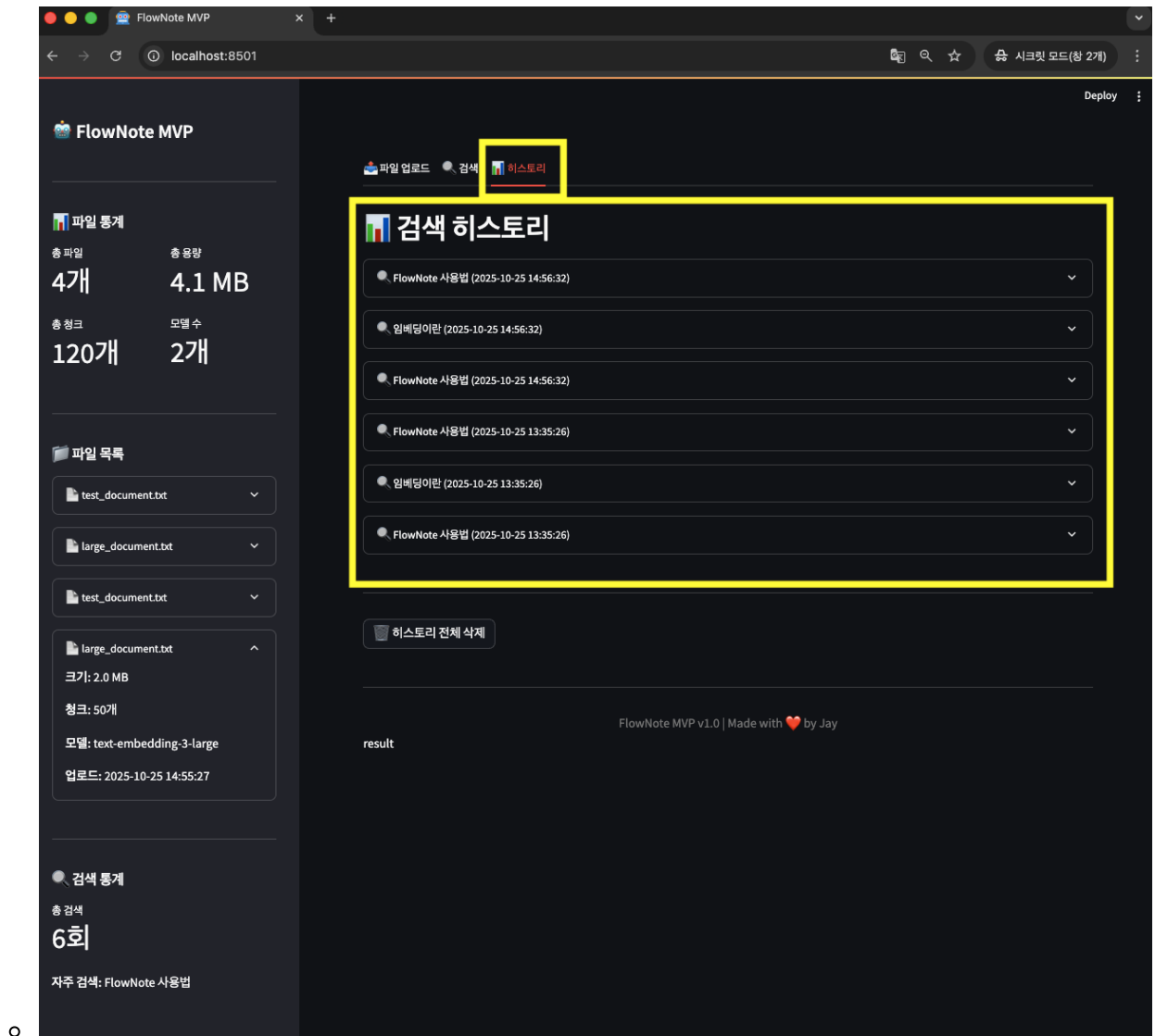




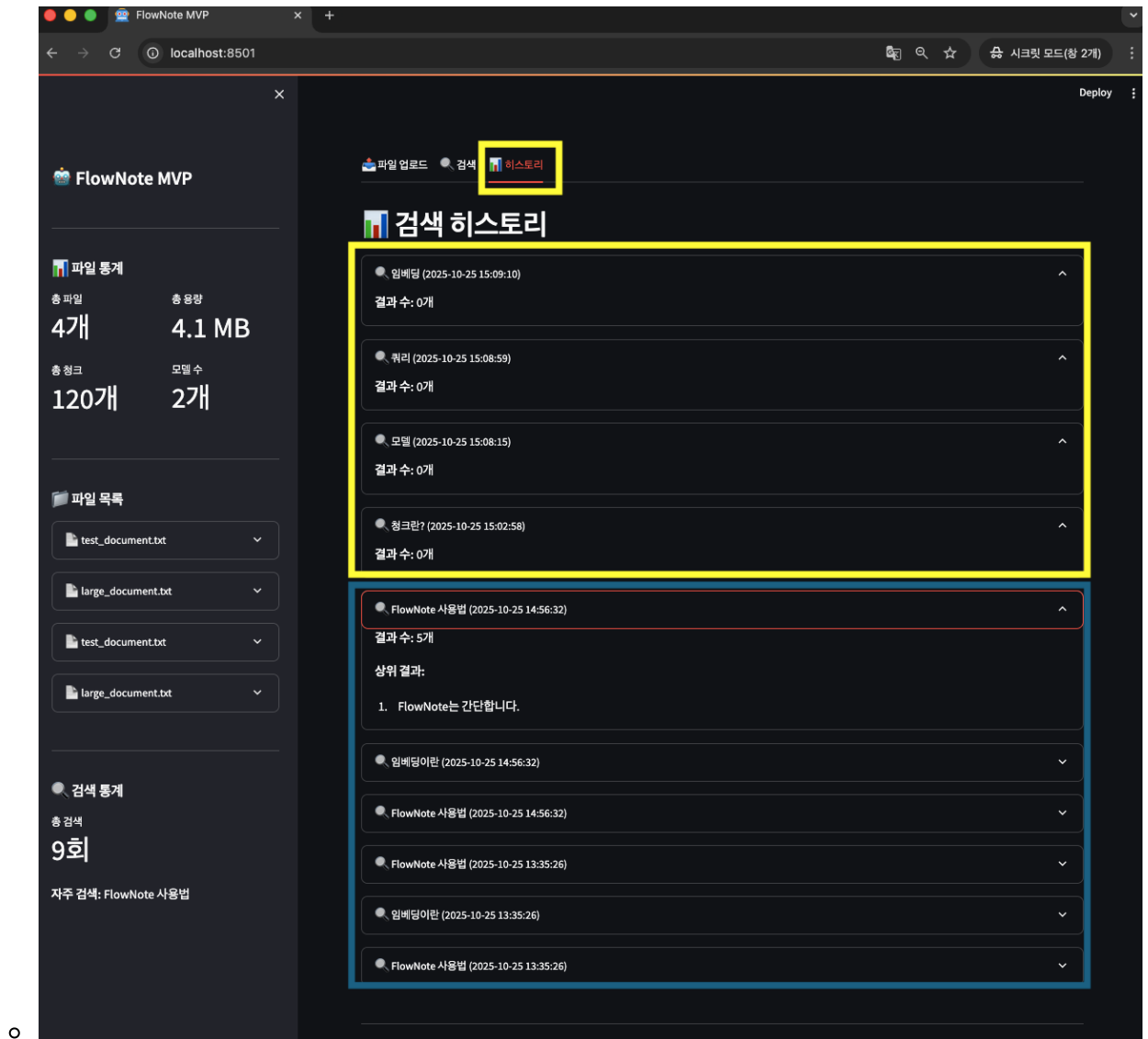
- ☒ 총 용량: 4.1 MB
- ☒ 총 청크: 120개
- ☒ 모델 수: 2개

㉓ 검색 히스토리 탭

- ☒ 검색 기록 표시

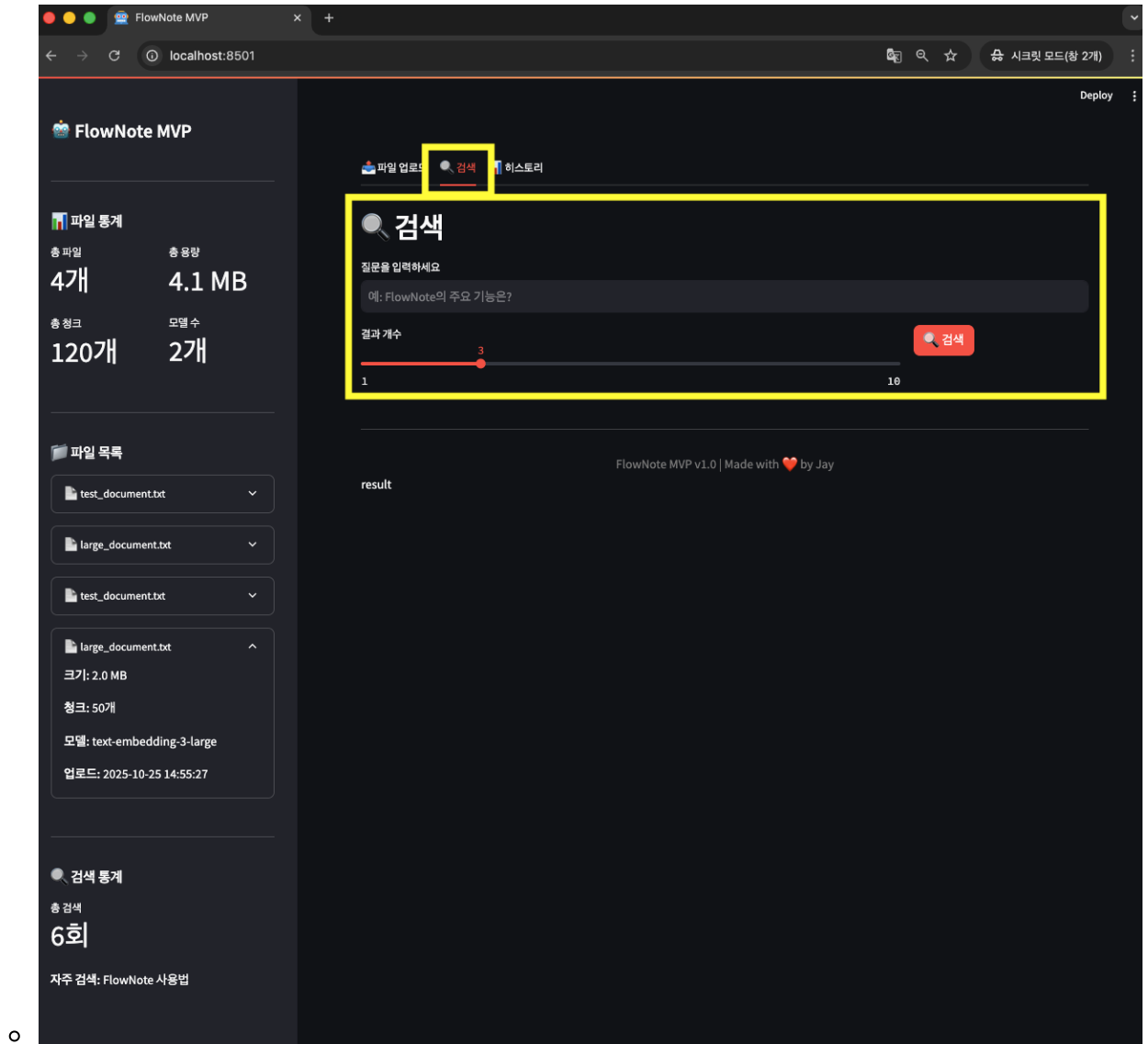


- ☒ 결과 수 표시
- ☒ 상위 결과 미리보기
- ☒ 0개 결과 발견 기록 ("0개 결과 발견" 메시지)

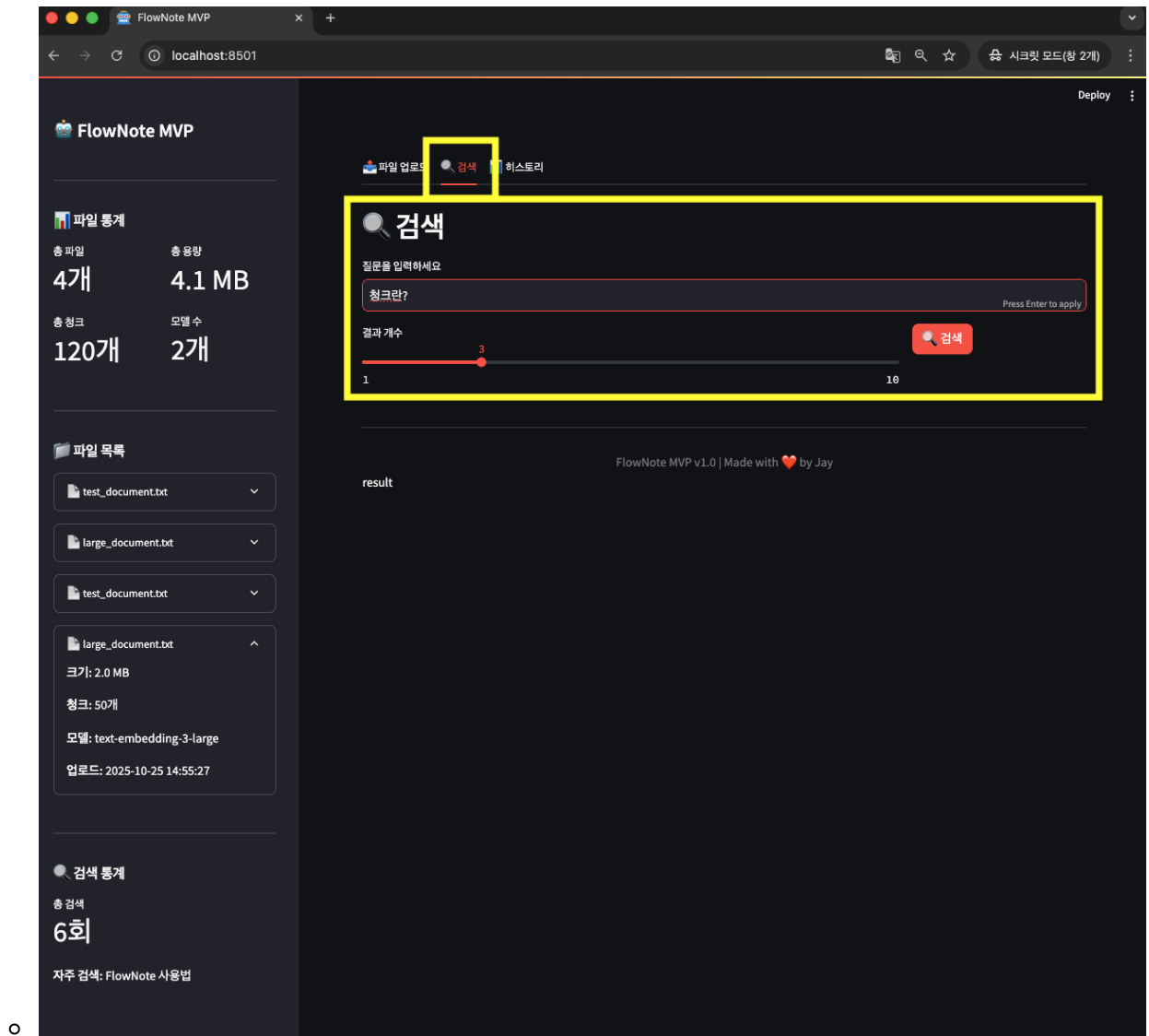


③ 검색 탭

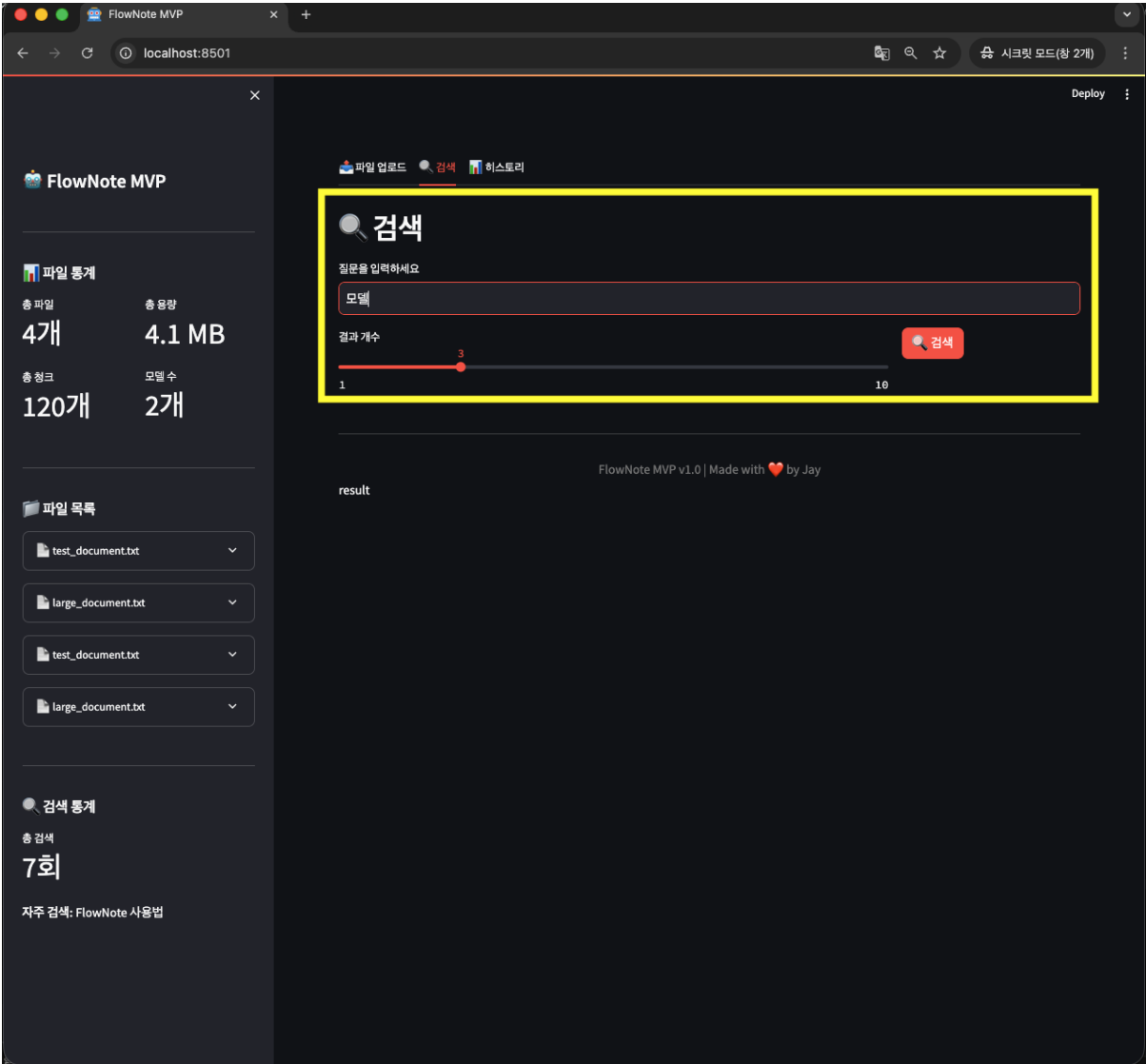
- ✓ 검색 입력창

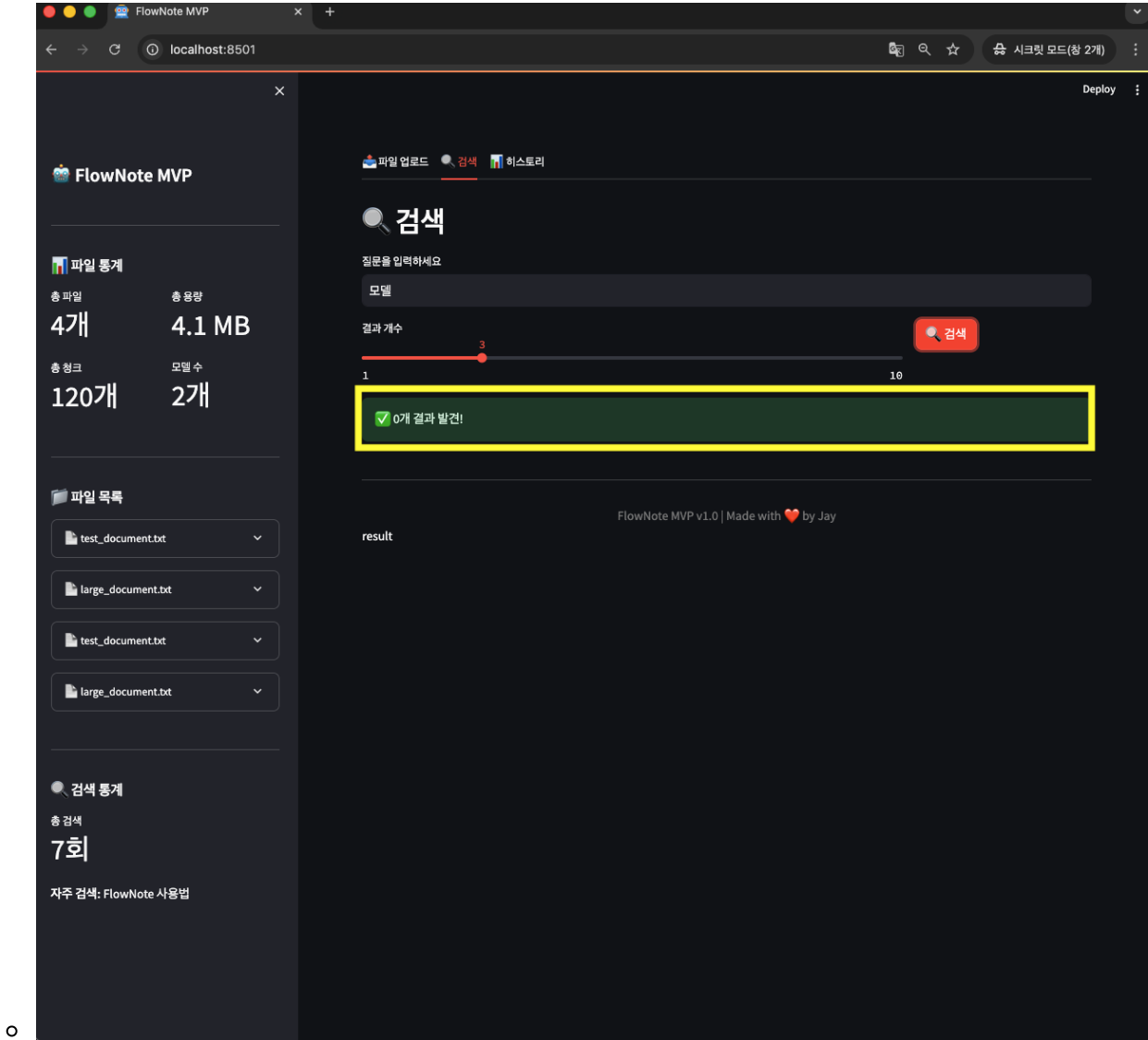


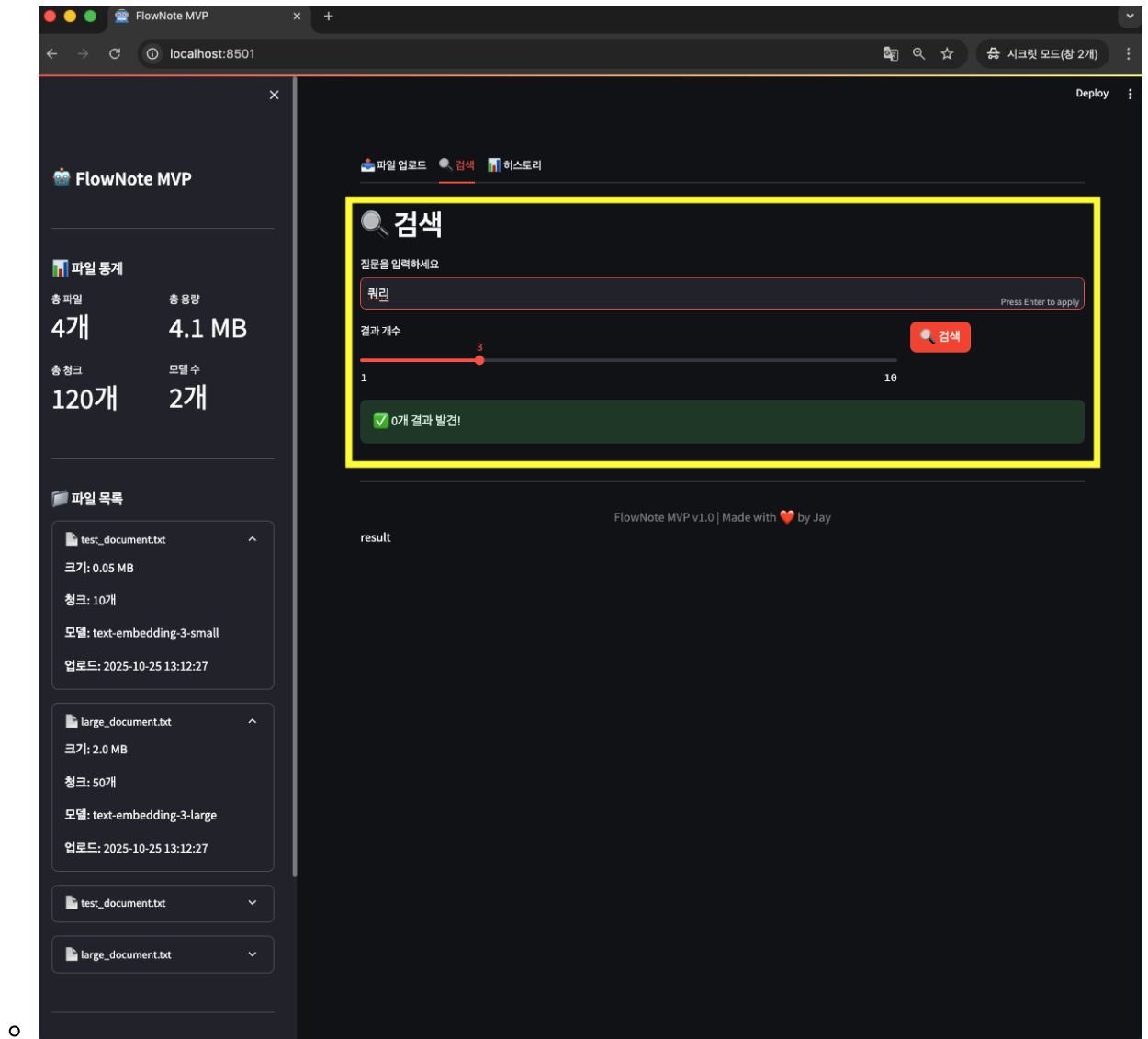
- ☒ 결과 개수 슬라이더
- ☒ 검색 버튼



- △ 테스트 데이터만 검색됨 (실제 파일 연동 안 됨)

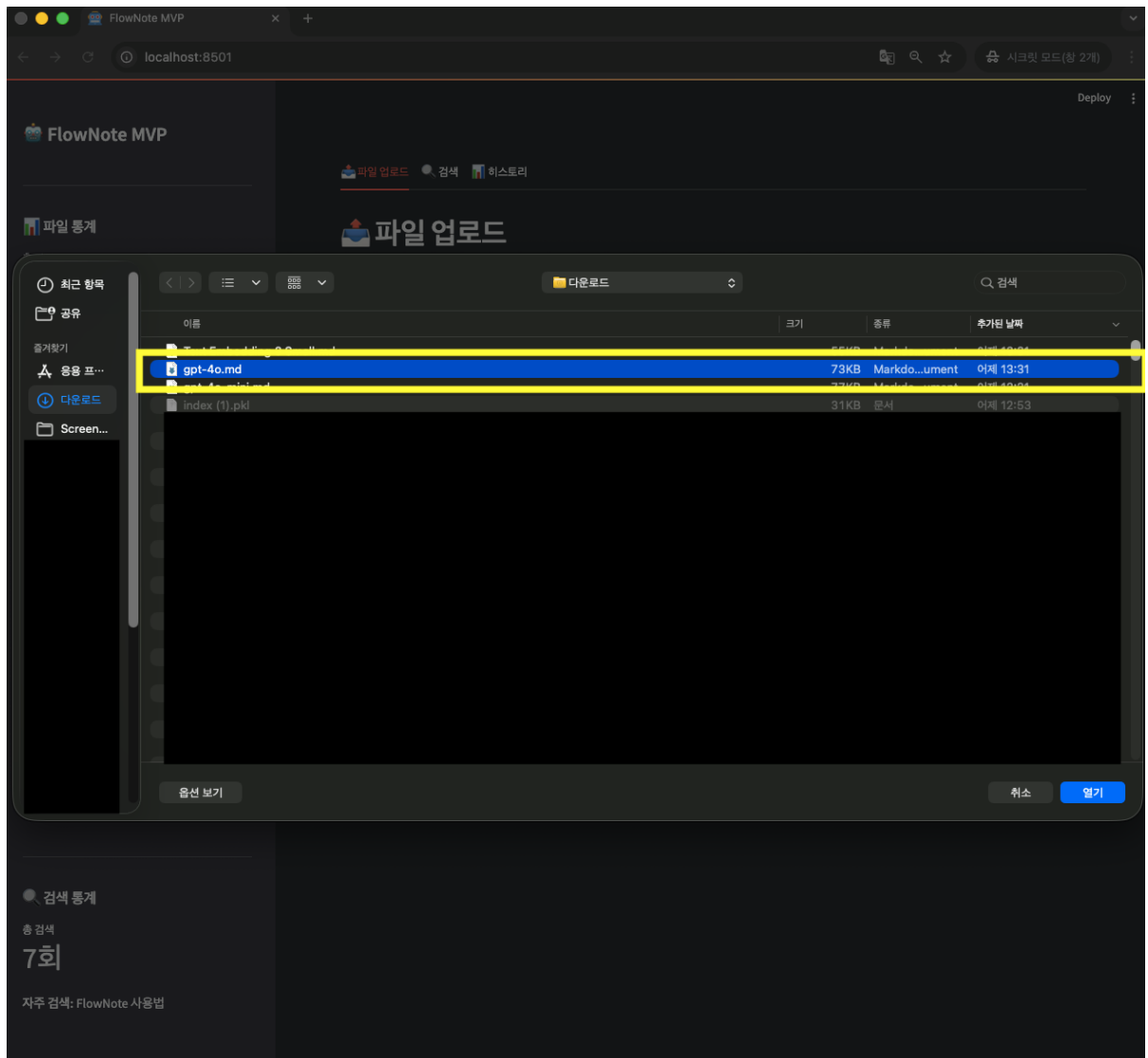




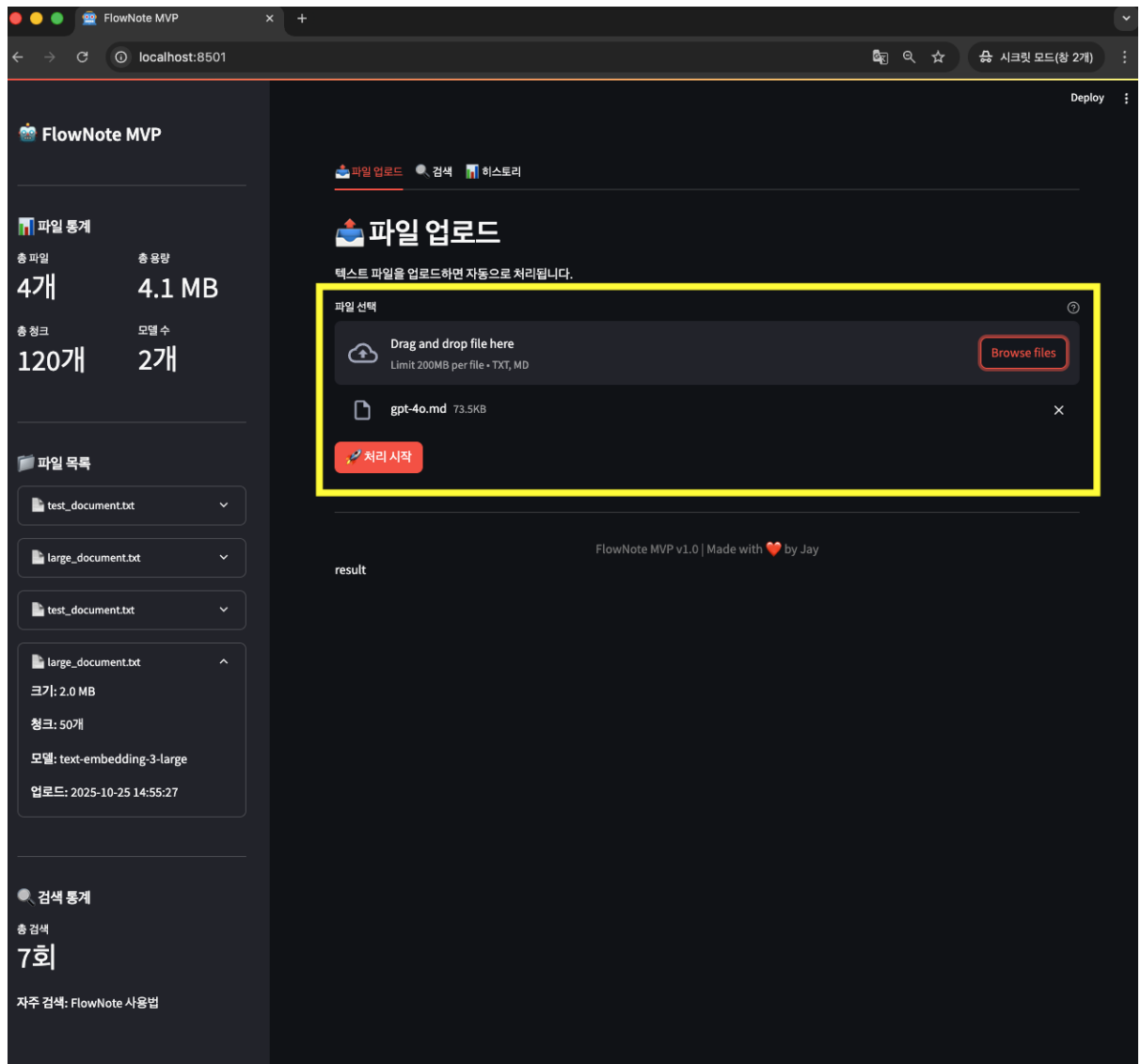


④ 파일 업로드 탭

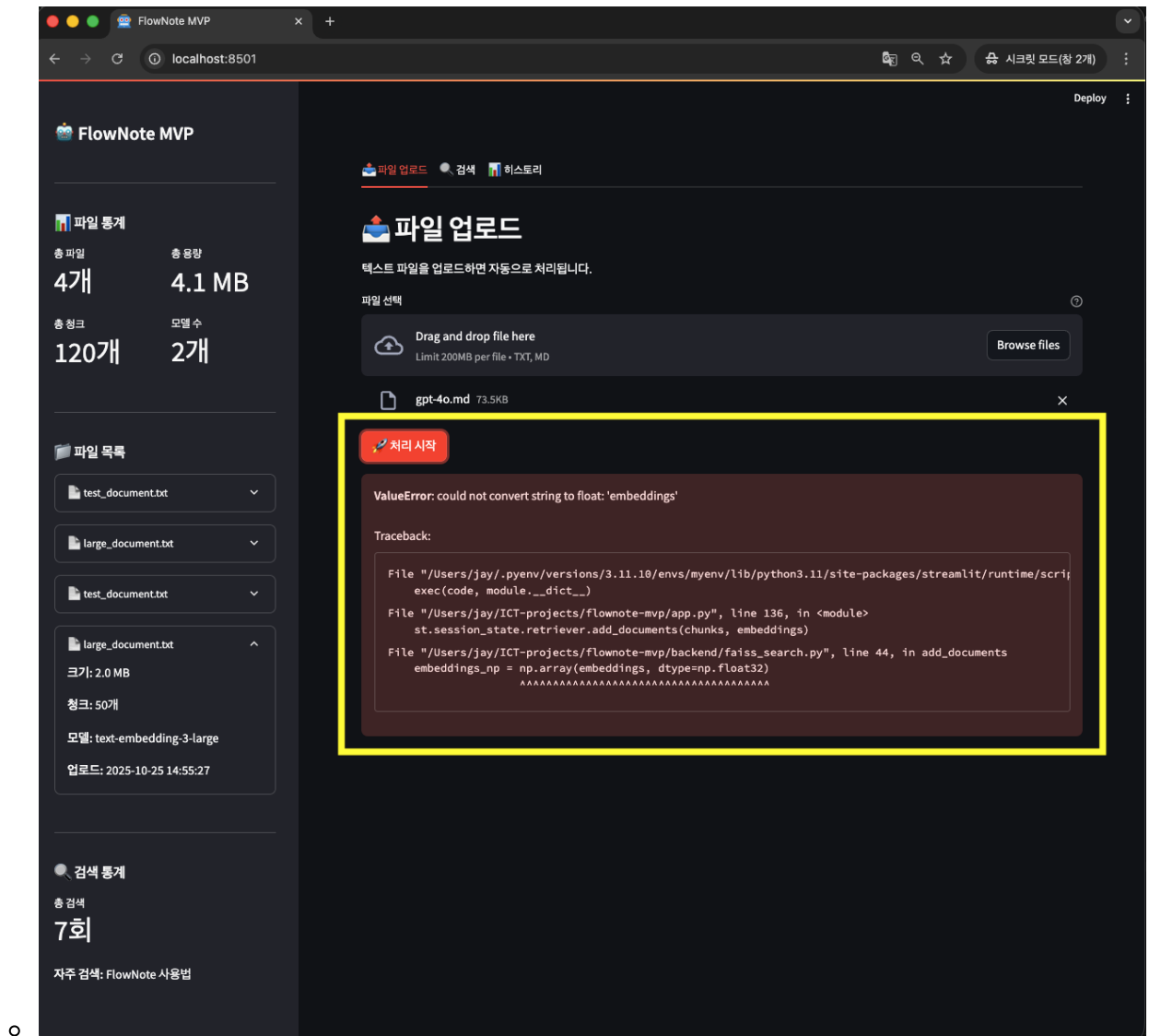
-  파일 업로드 UI



- ☒ 드래그 앤 드롭
- ☒ 파일 선택 버튼



- △ 업로드 후 처리 로직 없음



4. △ 발견된 이슈

1) 파일 업로드 → 검색 연동 미완성

- 현상:
 - 파일 업로드는 정상 작동
 - **data/uploads/** 폴더에 파일 저장됨
 - BUT 청킹/임베딩/FAISS 저장이 안 됨
 - 검색 시 테스트 데이터만 나옴
- 원인:
 - **app.py**에 파일 처리 로직이 빠짐
 - 업로드 후 청킹 → 임베딩 → FAISS 저장 흐름 없음
- 해결 필요:



2) FAISS 인덱스 영구 저장 미구현

- 현상:
 - 앱 재시작 시 인덱스 초기화됨
 - 이전 업로드 파일 검색 불가
- 해결 필요:
 - FAISS 인덱스를 `data/faiss/` 폴더에 저장
 - 앱 시작 시 인덱스 로드

5. 📊 테스트 요약

구성 요소	상태	비고
Backend 모듈	✅ 완료	모든 모듈 정상 작동
config.py	✅ 완료	API 설정 완료
utils.py	✅ 완료	유틸리티 함수 완료
chunking.py	✅ 완료	텍스트 분할 완료
embedding.py	✅ 완료	임베딩 생성 완료
faiss_search.py	✅ 완료	벡터 검색 완료
metadata.py	✅ 완료	메타데이터 관리 완료
search_history.py	✅ 완료	검색 기록 관리 완료
Streamlit UI	✅ 완료	UI 렌더링 완료
파일 업로드 → 검색	⚠ 미완성	연동 로직 필요

구성 요소	상태	비고
FAISS 영구 저장	⚠ 미완성	저장/로드 로직 필요

6. 🎯 다음 단계

1) 우선순위 1: 파일 처리 로직 추가

- ☐ `app.py`에 파일 업로드 후 처리 추가
- ☐ 청킹 → 임베딩 → FAISS 저장 흐름 구현

2) 우선순위 2: FAISS 인덱스 영구화

- ☐ FAISS 인덱스 저장 기능
- ☐ 앱 시작 시 인덱스 로드

3) 우선순위 3: 통합 테스트

- ☐ 전체 흐름 테스트
- ☐ 실제 파일로 검색 검증

7. 💡 결론

- 완료된 것:
 - ✅ Backend 핵심 모듈 8개 완성
 - ✅ Streamlit UI 완성
 - ✅ 개별 모듈 테스트 통과
- 남은 것:
 - ⚠ 파일 업로드 → 검색 연동 (Glue Code)
 - ⚠ FAISS 인덱스 영구 저장
- 진행률: **80% 완료** (Backend + UI 완성, 연동만 남음)

작성자: Jay
도움: Perplexity AI Assistant - *Claude-4.5-Sonnet*
다음 작업: 파일 처리 로직 구현 및 연동 완료