

FlowNote Dashboard - PARA Classifier

AI 기반 자동 분류 시스템

텍스트를 입력하면 AI가 PARA 방식으로 자동 분류하고, 신뢰도를 제시하는 대시보드



1. 📖 프로젝트 소개

FlowNote Dashboard는 당신의 할일/메모/자료를 **PARA 방식**(Projects, Areas, Resources, Archives)으로 자동 분류해주는 AI 기반 분류 시스템입니다.

💡 핵심 아이디어

당신의 입력 텍스트
↓
AI가 읽고 분석 (LangChain + GPT-4o)
↓
P/A/R/A 중 하나로 분류
↓
분류 결과 + 신뢰도 제시
↓
🎉 완료!

예시:

- 입력: "프로젝트 11월 30일까지 완료"
- 결과: 📁 Projects (신뢰도 100%)

2. 🚀 개발 히스토리 (Git 기반)

📊 Issue별 개발 진행도

Issue	단계	완료 날짜	핵심 기능	Git Commits	상태
#1	환경 구축	~10/23	- Python 3.11 환경 - 프로젝트 구조 - API 설정	#1.1 ~ #1.7	✅

Issue	단계	완료 날짜	핵심 기능	Git Commits	상태
#2	MVP v1.0	10/24-10/25	- Streamlit UI - 파일 업로드 - FAISS 검색 - 임베딩	#2.1 ~ #2.17	✅
#3	PARA 분류 v1	10/26-10/28	- PARA 분류기 - UI 통합 - 테스트	#3.1 ~ #3.3	✅
#4	Vision API	10/29	- GPT-4.1 API - Vision Helper - 모델 Config	#4.1 ~ #4.4	✅
#5	Dashboard v3.0	10/30-11/01	- LangChain 통합 - GPT-4o 분류 - React Frontend - FastAPI Backend	#5.1 ~ #5.3	🟦 진행 중

3. 📌 Issue별 상세 개발 과정

Issue #1: 환경 구축 ✅

- #1.1: FlowNote MVP 프로젝트 문서 초안
- #1.2: Python 3.11 개발 환경 구축
- #1.3: README.md 추가
- #1.4: API 환경 설정 & 테스트
- #1.5: .gitignore 업데이트 & Backend 구조
- #1.6: 프로젝트 구조 개선 & Assets 정리
- #1.7: 원격 잔여 이미지 정리

기술 스택:

- Python 3.11.10
- 프로젝트 구조 설계
- Git 환경 설정

Issue #2: MVP v1.0 - 기본 검색 시스템 ✅

- #2.1: Streamlit UI & 파일 업로드 기능
- #2.2: 청킹 & 임베딩 기능 추가
- #2.3: FAISS 검색 엔진 완성
- #2.4: Streamlit UI 검색 기능 완성
- #2.5: app.py 실습 문서 추가
- #2.6: Multi-Model API 설정
- #2.7: 파일 메타데이터 관리 기능

#2.8: 검색 히스토리 관리 기능
#2.9: Backend 모듈 통합 및 최적화
#2.10: Streamlit UI 개선 및 Backend 연동
#2.11: Backend 통합 테스트 결과 문서
#2.12: 파일 처리 로직 구현
#2.13: 사이드바 파일 목록 표시 기능
#2.14: MVP v1.0 공식 문서
#2.15: MVP 사용자 가이드
#2.16: 문서 업로드 및 검색 기능
#2.17: 마크다운 내보내기 기능

기술 스택:

- **Frontend:** Streamlit
- **검색:** FAISS (벡터 검색)
- **임베딩:** OpenAI text-embedding-3-small
- **파일 처리:** TXT, PDF (pdfplumber)
- **청킹:** RecursiveCharacterTextSplitter

Issue #3: PARA 분류 v1.0

#3.1: PARA 분류기 백엔드 구현
#3.2: PARA 분류 UI 통합 및 사이드바 개선
#3.3: PARA 분류기 테스트 파일 추가

기술 스택:

- PARA 분류 로직
- Streamlit UI 통합
- 테스트 파일

Issue #4: Vision API 통합

#4.1: GPT-4.1 API connection test
#4.2: config.py 클래스 기반 리팩토링
#4.3: ModelConfig 클래스 기반 통합 테스트
#4.4: Vision Helper 모듈 구현 & 통합 테스트

기술 스택:

- GPT-4.1 Vision API
 - ModelConfig 클래스
 - Vision Helper 모듈
-

Issue #5: Dashboard v3.0 - AI 자동 분류 🟦 진행 중

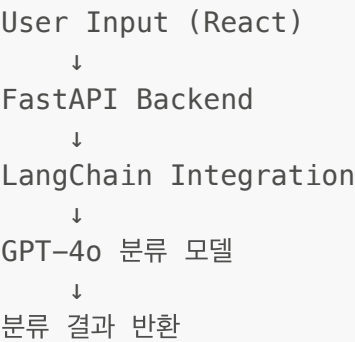
```
#5.1: DatabaseConnection class + SQLite schema
#5.2: MetadataAggregator core logic
#5.3: Dashboard UI structure
#5.4: update .gitignore
#5.5: .gitignore에 Streamlit 설정 추가

# PARA 분류 고도화
#5.2.1: AI PARA classifier module
#5.2.2: LangChain + GPT 통합 PARA 분류기
#5.2.3: ParaClassifier 컴포넌트 추가 (React)
```

기술 스택:

- **Backend:** FastAPI
- **Frontend:** React 18
- **AI 통합:** LangChain
- **LLM:** OpenAI GPT-4o
- **Database:** SQLite (예정)
- **분류 시스템:** PARA Method

현재 구조:



4. 🏗️ 기술 스택 (분야별)

4.1 🧩 Backend

기술	버전	용도	도입 Issue
Python	3.11.10	개발 언어	#1
FastAPI	latest	REST API 프레임워크	#5
LangChain	>=0.1.0	AI 체인 및 프롬프트 관리	#5
SQLite	3	메타데이터 저장소	#5

기술	버전	용도	도입 Issue
Uvicorn	0.24.0	ASGI 서버	#5

4.2 🎨 Frontend

기술	버전	용도	도입 Issue
React	18+	UI 라이브러리	#5
JavaScript	ES6	개발 언어	#5
CSS	3	스타일링	#5
Streamlit	latest	초기 UI (v1-v2)	#2

4.3 🧠 LLM & AI


기술	모델	용도	도입 Issue
OpenAI API	GPT-4o	PARA 분류	#5
OpenAI API	GPT-4.1 Vision	이미지 분석 (예정)	#4
OpenAI Embeddings	text-embedding-3-small	벡터 임베딩	#2

4.4 🔍 검색 & 데이터


기술	버전	용도	도입 Issue
FAISS	latest	벡터 검색 엔진	#2
pdfplumber	latest	PDF 파싱	#2
python-dotenv	1.0.0	환경변수 관리	#1

5. ✨ 핵심 기능 (v3.0)


5.1 AI 기반 자동 분류

 Projects (프로젝트)

→ 기한/마감일이 있는 구체적 목표
예: "11월까지 프로젝트 완료"

 Areas (분야)

→ 지속적 책임 영역
예: "팀 성과 평가는 계속 진행 업무"

 Resources (자료)

→ 참고용 정보/자료
예: "API 사용 가이드"



Archives (보관)

→ 완료되고 참고용이 된 것

예: "작년 프로젝트 결과"

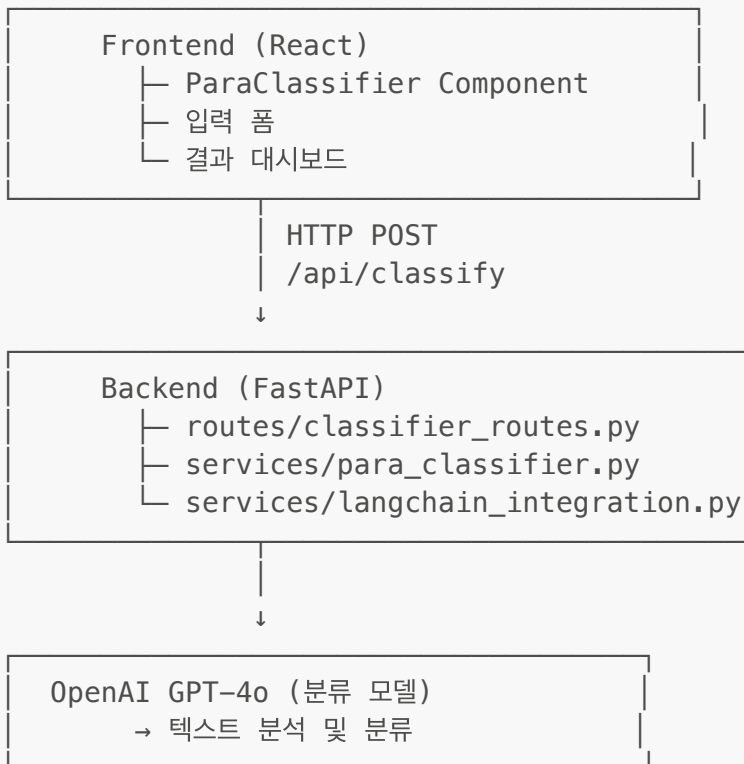
5.2 신뢰도 점수

- 각 분류의 신뢰도를 **0-100%**로 표시
- AI의 판단 근거 설명 제공
- 감지된 신호 표시

5.3 직관적인 대시보드

- **React** 기반 모던 UI
- 실시간 분류 결과 표시
- 분류 히스토리 관리

6. 🏗️ 아키텍처



7. 📁 프로젝트 구조

```
flownote-mvp/  
├─ requirements.txt  
# Python 의존성 (Root)
```

```

├── .env.example                                # 환경변수 예시
├── .gitignore
├── backend/                                   # FastAPI 백엔드
│   ├── app.py                                # 메인 앱
│   ├── routes/
│   │   └── classifier_routes.py              # /api/classify 엔드포인트
│   ├── services/
│   │   ├── para_classifier.py                # PARA 분류 로직
│   │   └── langchain_integration.py          # LangChain 통합
│   └── prompts/
│       └── para_classification_prompt.txt     # 분류 프롬프트
├── frontend/                                 # React 프론트엔드
│   ├── public/
│   ├── src/
│   │   ├── components/
│   │   │   ├── ParaClassifier.jsx           # 메인 컴포넌트
│   │   │   └── ParaClassifier.css           # 스타일
│   │   ├── App.js
│   │   ├── App.css
│   │   └── index.js
│   ├── package.json
│   └── .env (자동 설정)
├── data/                                     # 데이터 저장소
├── docs/                                     # 문서
│   ├── constitution.md                      # 프로젝트 헌장
│   └── practices/                           # Phase 1에서 backend 파일들 확장 중 과정 기록 폴
├── specs/                                   # 기능 명세서
│   ├── file-upload.md                      # 파일 업로드 관련
│   ├── faiss-search.md                    # 검색 관련
│   └── prompt-templates.md                 # 결과 변환 관련 *result_markdown
├── markdown-export.md                       # 결과 저장 관련 *(data/exports/)*
├── mcp-classification.md                   # 업로드된 파일 자동 분류 관련
├── troubleshooting/                         # 트러블슈팅 기록 문서 보관
│   ├── PromptTemplate_Escaping.md          # PromptTemplate 중괄호 이
│   └── PromptTemplate_Escaping.pdf          스케이프 문제 해결 가이드
├── assets/                                  # 프로젝트에서 사용되는 첨부파일 및 사각자료 등 정적 자원 포함하
│   ├── images/                             # 일반 이미지 (스크린샷, 배경 등)
│   └── figures/                             # 문서에 포함되는 분석 결과 이미지

```

```
|— app.py                # 스트림릿 메인 실행 앱
|— app_classifier.py     # 스트림릿 - 분류 페이지 앱
|— USER_GUIDE.md        # 유저 가이드
|— README.md             # 해당 파일
```

8. 🚀 빠른 시작

8.1 사전 요구사항

- **Python:** 3.11+
- **Node.js:** 18+
- **OpenAI API Key:** platform.openai.com

8.2 설치 방법

8.2.1 Backend 설정

```
# 1. 리포지토리 클론
git clone https://github.com/jjaayy2222/flownote-dashboard.git
cd flownote-dashboard

# 2. 가상환경 생성
python -m venv venv
source venv/bin/activate # Windows: venv\Scripts\activate

# 3. 패키지 설치 (Root에서)
pip install -r requirements.txt

# 4. 환경변수 설정
cp .env.example .env
# .env 파일에 OPENAI_API_KEY 입력

# 5. Backend 실행
cd backend
python app.py
# → http://127.0.0.1:8000 에서 실행됨
```

8.2.2 Frontend 설정

```
# 1. Frontend 폴더로 이동
cd frontend

# 2. 패키지 설치
npm install
```



```
# 3. Frontend 실행
npm start
# → http://localhost:3000 에서 자동 열림
```

9. 📖 사용 방법


9.1 Step 1: 텍스트 입력

예1: "프로젝트 11월 30일까지 완료"
예2: "팀 성과 평가는 계속 진행해야 해"
예3: "API 사용 가이드 문서"

9.2 Step 2: 분류 실행

- "분류하기" 버튼 클릭

9.3 Step 3: 결과 확인

 분류 결과
카테고리: Projects
신뢰도: 100.0%
설명: 기한 표현(11월 30일까지)과 구체적 목표(프로젝트 완료)가 있어 Projects로 분류됨.
감지된 신호: 11월 30일까지, 프로젝트, 완료

10. 🎯 PARA 방식 설명

10.1 📅 P - Projects (프로젝트)

- 정의: 구체적인 마감일이 있는 목표
- 특징: 기한, 마일스톤, 구체적 목표
- 예시:
 - "11월 5일까지 프로젝트 완료"
 - "Dashboard 구현 (11/30 완료)"

10.2 🌐 A - Areas (분야)

- 정의: 지속적으로 관리하는 책임 영역
- 특징: 진행 상황 모니터링, 지속적 책임
- 예시:
 - "팀 성과 평가는 계속 진행해야 해"
 - "직원 관리 및 발전 (지속)"

10.3 📚 R - Resources (자료)

- 정의: 참고/활용용 정보 및 자료
- 특징: 정적 정보, 참고용
- 예시:
 - "API 사용 가이드"
 - "Python 최적화 팁"

10.4 📦 A - Archives (보관)

- 정의: 완료되고 보관만 하는 것
- 특징: 참고용, 완료됨
- 예시:
 - "작년 프로젝트 정리"
 - "2024년 회의록"

11. 🧪 테스트

11.1 API 테스트 (Backend)

```
cd backend

# API 직접 테스트
curl -X POST http://localhost:8000/api/classify \
  -H "Content-Type: application/json" \
  -d '{
    "text": "프로젝트 11월 30일까지 완료",
    "filename": "test.txt"
  }'

# 예상 응답:
# {
#   "result": {
#     "category": "Projects",
#     "confidence": 1.0,
#     "reasoning": "...",
#     "signals": ["11월 30일까지", "프로젝트", "완료"]
#   }
# }
```

11.2 Frontend 테스트 (콘솔)

```
// 브라우저 개발자 도구 (F12) → Console 탭

async function testAPI() {
  const response = await fetch('http://localhost:8000/api/classify', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({
      text: '팀 성과 평가는 계속 진행하는 업무야',
    })
  })
}
```

```
        filename: 'test.txt'
    })
});

const data = await response.json();
console.log(data);
}

testAPI();
```

12. 🗺️ 개발 로드맵

✅ Issue #1: 환경 구축 (완료)

- ☒ Python 3.11 환경
- ☒ 프로젝트 구조 설계
- ☒ Git 설정
- ☒ API 환경 설정

✅ Issue #2: MVP v1.0 (완료)

- ☒ Streamlit UI
- ☒ 파일 업로드 (TXT, PDF)
- ☒ FAISS 검색 엔진
- ☒ OpenAI 임베딩
- ☒ 검색 히스토리
- ☒ 마크다운 내보내기

✅ Issue #3: PARA 분류 v1.0 (완료)

- ☒ PARA 분류기 백엔드
- ☒ UI 통합
- ☒ 테스트 파일

✅ Issue #4: Vision API (완료)

- ☒ GPT-4.1 API 연결
- ☒ ModelConfig 리팩토링
- ☒ Vision Helper 모듈

🟡 Issue #5: Dashboard v3.0 (진행 중, ~11/12)

- ☒ SQLite 데이터베이스
- ☒ MetadataAggregator
- ☒ Dashboard UI 구조
- ☒ AI PARA 분류 모듈
- ☒ LangChain + GPT-4o 통합
- ☒ React Frontend

- ☒ FastAPI Backend
- ☐ 분류 정확도 개선
- ☐ 에러 처리 강화
- ☐ 배치 처리 기능

🚧 Issue #6: 고급 기능 (예정, ~11월 말)

- ☐ LangGraph 멀티 스텝 분류
- ☐ 태그 자동 생성
- ☐ 유사 문서 찾기
- ☐ 분류 히스토리 저장
- ☐ 통계 대시보드

🧙 Issue #7: 통합 시스템 (예정, 12월 이후)

- ☐ Notion 연동
- ☐ Obsidian Export
- ☐ 자동 폴더 구조 생성
- ☐ 배포 (Railway/Vercel)

13. 📊 Issue별 진행 상태

Issue	진행률	완료 항목	남은 항목	상태
#1	100%	7/7	0/7	✅ 완료
#2	100%	17/17	0/17	✅ 완료
#3	100%	3/3	0/3	✅ 완료
#4	100%	4/4	0/4	✅ 완료
#5	80%	8/10	2/10	🟦 진행 중
#6	0%	0/5	5/5	⌚ 대기
#7	0%	0/4	4/4	⌚ 대기

14. ❓ FAQ

Q1. Backend와 Frontend를 동시에 실행해야 하나요?

- A: 네, 둘 다 실행해야 합니다.
 - Backend: `python app.py` (포트 8000)
 - Frontend: `npm start` (포트 3000)

Q2. API Key를 어디서 얻나요?

- A: [OpenAI Platform](#)에서 발급 가능합니다.

Q3. 무료로 사용할 수 있나요?

- **A:** 무료 사용량 내에서 가능합니다. (초기 \$5 크레딧 제공)

Q4. 데이터는 어디에 저장되나요?

- **A:** 모든 데이터는 **현재 로컬**에만 저장됩니다. (OpenAI API 제외)

Q5. 다른 AI 모델을 사용할 수 있나요?

- **A:** `backend/services/langchain_integration.py`를 수정하면 *Claude* 등 다른 모델 사용 가능합니다.

Q6. requirements.txt 위치가 궁금해요

- **A:** **Root** 디렉토리에 있습니다. (`flownote-dashboard/requirements.txt`)

15. 🤝 기여하기

15.1 이슈 제보

- 버그: [GitHub Issues](#)
- 기능 제안: [Discussions](#)

15.2 기여 방법

- **Fork** 후 새 브랜치 생성
- 변경 후 커밋 (커밋 메시지 형식: `feat [#이슈번호]: 설명`)
- **Pull Request** 제출

16. 📄 라이선스

MIT License - 자유롭게 사용, 수정, 배포 가능합니다.

17. 👤 개발자

Jay

- GitHub: [@jjaayy2222](#)

18. 🙏 감사의 말

이 프로젝트는 다음 기술/도구 덕분에 가능했습니다:

- **OpenAI** - GPT-4o 모델
- **LangChain** - AI 체인 프레임워크
- **FastAPI** - 빠른 API 개발
- **React** - 모던 UI
- **Perplexity AI** - 개발 조력
- **Claude & bomi** 멘토님 - 멘토링 & 검수

FlowNote Dashboard - AI가 당신의 할일을 정리해줍니다 🚀

Made with ❤️ by [Jay](#)
