

# FlowNote 사용자 가이드

FlowNote로 문서를 AI 기반으로 자동 분류하고 관리하는 방법을 안내합니다.

## 목차

1. 시작하기
2. 온보딩 가이드
3. 파일 분류
4. 키워드 검색
5. 대시보드 활용
6. 고급 기능
7. 문제 해결
8. 팁 & 트릭

## 1. 시작하기

### 1.1 환경 요구사항

- **Python:** 3.11 이상
- **운영체제:** Windows, macOS, Linux
- **OpenAI API Key:** [platform.openai.com](https://platform.openai.com)
- **필수 라이브러리:** [requirements.txt](#) 참조

### 1.2 설치 및 실행

#### Step 1: 저장소 클론

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

#### Step 2: 가상환경 설정

```
# Windows
python -m venv venv
venv\Scripts\activate

# macOS/Linux
python3 -m venv venv
source venv/bin/activate
```

#### Step 3: 패키지 설치

```
pip install -r requirements.txt
```

#### Step 4: 환경 변수 설정

프로젝트 루트에 `.env` 파일 생성:

```
# OpenAI API Keys
OPENAI_API_KEY=sk-your-api-key-here

# GPT-4o (분류 전용)
GPT4O_API_KEY=sk-your-api-key-here
GPT4O_BASE_URL=https://api.openai.com/v1
GPT4O_MODEL=gpt-4o

# GPT-4o-mini (경량 작업)
GPT4O_MINI_API_KEY=sk-your-api-key-here
GPT4O_MINI_BASE_URL=https://api.openai.com/v1
GPT4O_MINI_MODEL=gpt-4o-mini

# Embeddings
EMBEDDING_API_KEY=sk-your-api-key-here
EMBEDDING_BASE_URL=https://api.openai.com/v1
EMBEDDING_MODEL=text-embedding-3-small
```

#### Step 5: 앱 실행

**Backend (FastAPI) 실행:**

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

**Frontend (Streamlit) 실행 (새 터미널):**

```
cd streamlit
streamlit run app.py
# → http://localhost:8501 에서 자동 실행
```

**대시보드 실행 (선택사항, 새 터미널):**

```
streamlit run streamlit/pages/dashboard.py
# → http://localhost:8502 에서 실행
```

## Step 6: 브라우저에서 확인

- 자동으로 브라우저가 열립니다
  - 열리지 않으면 터미널의 URL을 직접 복사하여 접속
- 

## 2. 🚀 온보딩 가이드

### 2.1 온보딩이란?

온보딩은 FlowNote가 당신의 직업과 관심 영역을 학습하여 **맞춤형 분류**를 제공하기 위한 초기 설정 과정입니다.

### 2.2 온보딩 단계


#### Step 1: 기본 정보 입력

1. **Tab 1: 온보딩** 클릭
2. **이름** 입력 (예: **Jay**)
3. **직업** 입력 (예: **개발자**, **디자이너**, **교사**)
4. **다음 단계** → 버튼 클릭
5. GPT-4o가 당신의 직업에 맞는 **10개 영역** 추천


#### Step 2: 관심 영역 선택

1. 추천된 10개 영역 중 **정확히 5개** 선택
2. 선택 예시:
  - **Python Development**
  - **Machine Learning**
  - **Web Development**
  - **Data Science**
  - **Project Management**
3. **완료** → 버튼 클릭

#### Step 3: 완료 확인

-  온보딩 완료!
- 사용자 정보 확인:
  - 이름
  - 직업
  - User ID
  - 선택한 5개 영역

### 2.3 온보딩 재설정

- 온보딩 완료 후  **온보딩 다시하기** 버튼으로 초기화 가능
  - 직업이나 관심 영역이 변경되었을 때 재설정 권장
- 

## 3. 📄 파일 분류

### 3.1 파일 업로드

#### 단계별 안내

1. **Tab 2: 파일 분류 이동**
2. 온보딩 완료 여부 확인 (미완료 시 경고 메시지)
3. 🗂️ **분류할 파일 업로드 클릭**
4. 파일 선택 (PDF, TXT, MD 지원)
5. 파일 정보 확인:
  - 파일명
  - 파일 크기
  - 파일 타입

#### 지원 파일

- ✅ PDF (.pdf)
- ✅ TXT (.txt)
- ✅ Markdown (.md)

### 3.2 자동 분류 실행

1. 🚀 **분류 시작** 버튼 클릭
2. AI 분석 진행 중... (사용자 맥락 반영)
3. 분류 결과 확인:
  - **카테고리:** Projects/Areas/Resources/Archives
  - **신뢰도:** 0-100%
  - **맥락 반영:** ✅/❌
  - **키워드 수:** 추출된 키워드 개수

### 3.3 분류 결과 이해하기

#### PARA 카테고리 설명


- 📁 **Projects (프로젝트)**
  - 기한/마감일이 있는 구체적 목표
  - 예: "11월 30일까지 대시보드 구현"
  - 신호: "마감일", "프로젝트", "완료", "목표"
- 🎯 **Areas (분야)**
  - 지속적 책임 영역
  - 예: "팀 성과 관리는 계속 진행"
  - 신호: "지속", "관리", "모니터링", "유지"
- 📖 **Resources (자료)**
  - 참고용 정보/학습 자료
  - 예: "Python 최적화 가이드"
  - 신호: "가이드", "참고", "문서", "학습"
- 📦 **Archives (보관)**

- 완료된 프로젝트 보관
- 예: "2024년 프로젝트 결과"
- 신호: "완료", "종료", "보관", "과거"

## 신뢰도 점수 해석





- **90-100%**: 매우 확실한 분류
- **70-89%**: 높은 신뢰도
- **50-69%**: 중간 신뢰도 (수동 확인 권장)
- **50% 미만**: 낮은 신뢰도 (재분류 권장)

## 3.4 분류 히스토리


- 사이드바  **분류 히스토리**에서 확인
- 최근 5개 분류 결과 표시
- **초기화** 버튼으로 히스토리 삭제

## 4. 🔍 키워드 검색


### 4.1 문서 업로드 및 처리

1. **Tab 3: 키워드 검색** 이동
2. 여러 문서 업로드 (다중 선택 가능)
3.  **파일 처리** 버튼 클릭
4. 처리 과정 확인:
  -  텍스트 분석 중...
  -  임베딩 생성 중...
  -  검색 인덱스 구축 중...

### 4.2 검색 실행

1. 검색어 입력 (예: **프로젝트 목표**)
2. 검색 결과 개수 선택 (슬라이더, 1-10개)
3.  **키워드 검색** 버튼 클릭

### 4.3 검색 결과 확인

 검색 결과 (3개)



 결과 #1 | example.pdf | PDF | 점수: 0.8234

프로젝트의 주요 목표는 11월 30일까지 대시보드 구현을 완료하는 것입니다...

키워드: 프로젝트, 목표, 마감일, 대시보드, 구현

신뢰도: 95%

## 4.4 검색 결과 내보내기





1.  검색 결과 MD로 내보내기 버튼 클릭
2.  다운로드 버튼으로 마크다운 파일 저장
3. 파일명: `flownote_search_YYYYMMDD_HHMMSS.md`

## 5. 대시보드 활용

### 5.1 분류 통계 확인

Tab 4: 분류 통계 또는 Dashboard 페이지에서 확인

#### KPI 메트릭

-  전체 파일: 총 파일 개수 (변화량 표시)
-  총 검색: 총 검색 횟수 (변화율 표시)
-  분류율: 분류 완료율
-  평균 신뢰도: 평균 분류 신뢰도

#### PARA 분포

- Projects/Areas/Resources/Archives별 파일 개수
- 바 차트로 시각화

### 5.2 메타데이터 확인

Tab 5: 메타데이터에서 확인

#### 세션 데이터

- 이번 세션 분류 목록
- 사용자 필터링 가능
- 파일명, 카테고리, 신뢰도, 시간 표시

#### 전체 DB 데이터

- 모든 분류 기록 확인
- 충돌 플래그 표시
- Snapshot ID 확인

### 5.3 파일 트리

Dashboard 페이지 →  파일 트리

- 계층 구조로 파일 표시

- 최대 7개 표시 (나머지는 expander)
- 📁 폴더, 📄 파일 아이콘 구분

## 5.4 최근 활동 로그

Dashboard 페이지 → 🔥 최근 활동 탭

- 최근 10개 분류 활동 표시
- 타임스탬프, 파일명, 카테고리, 신뢰도 확인

---

## 6. 🎯 고급 기능

### 6.1 맥락 기반 분류

맥락 반영이란?

- 사용자의 직업과 관심 영역을 고려한 분류
- 동일한 문서도 사용자에게 따라 다르게 분류될 수 있음

예시:

문서: "Python 최적화 가이드"

개발자 → Resources (자료)

학생 → Areas (학습 분야)

강사 → Projects (강의 준비)

### 6.2 배치 처리 (예정)

- 여러 파일 동시 처리
- 폴더 단위 업로드
- 자동 분류 및 정리

### 6.3 태그 자동 생성 (예정)

- AI가 키워드 태그 자동 생성
- 태그 기반 검색
- 태그 클라우드 시각화

---

## 7. 🛠 문제 해결

### 7.1 자주 발생하는 오류

❌ OPENAI\_API\_KEY not found

원인:

- .env 파일 누락

- 환경 변수 설정 오류

해결:

```
# .env 파일 생성
echo "OPENAI_API_KEY=sk-your-key-here" > .env

# 앱 재시작
streamlit run streamlit/app.py
```

### ❌ 온보딩 필요 경고

원인:

- 온보딩을 완료하지 않음

해결:

- Tab 1: 온보딩에서 온보딩 완료

### ❌ API 오류: 401 Unauthorized

원인:

- API Key 잘못됨
- API Key 만료

해결:

- .env 파일의 API Key 확인
- 새 API Key 발급 후 재설정

### ❌ 파일 업로드 실패

원인:

- 파일 형식 불일치 (PDF/TXT/MD만 지원)
- 파일 손상
- 메모리 부족

해결:

1. 파일 형식 확인
2. 파일 무결성 확인 (다른 앱에서 열어보기)
3. 파일 크기 확인 (권장: 10MB 이하)

### ❌ 검색 결과 없음

원인:



- 검색어와 문서 내용 불일치
- 파일 처리 미완료

#### 해결:

1. 검색어를 더 일반적으로 변경
2. K 값 증가 (더 많은 결과 요청)
3. 파일 목록에서 처리 완료 확인

## 7.2 성능 최적화 팁

### 파일 전처리

- 불필요한 페이지 제거
- 텍스트 품질 확인
- 이미지는 OCR 처리

### 검색 최적화

- 명확한 키워드 사용
- 너무 긴 질문 피하기
- 핵심 단어 포함

### 리소스 관리

- 불필요한 파일 삭제
- 주기적으로 캐시 정리
- 메모리 사용량 모니터링

---

## 8. 💡 팁 & 트릭

### 8.1 효율적인 검색법

❌ 나쁜 예:  
"이 프로젝트에서 우리가 달성하고자 하는 목표가 무엇인지 자세히 알려주세요"

✅ 좋은 예:  
"프로젝트 목표"

### 8.2 파일명 규칙

- 명확한 이름 사용
- 날짜 포함 (예: 2025-11-project-report.pdf)
- 버전 표시 (예: budget\_v2.pdf)

### 8.3 분류 정확도 향상

### 1. 온보딩 정보를 정확하게 입력

- 직업을 구체적으로 작성
- 관심 영역을 신중하게 선택

### 2. 파일명을 명확하게 작성

- 내용을 유추할 수 있는 파일명 사용
- 카테고리를 암시하는 키워드 포함

### 3. 신뢰도가 낮으면 재분류

- 50% 미만 신뢰도는 재분류 권장
- 파일 내용 보완 후 재업로드

## 8.4 생산성 향상 워크플로우

1. 온보딩 완료 (최초 1회)
2. 파일 일괄 업로드
3. 자동 분류 실행
4. 신뢰도 낮은 파일 재확인
5. 키워드 검색으로 상세 탐색
6. 대시보드에서 전체 현황 확인
7. 필요 시 마크다운 내보내기

## 8.5 통합 활용 (예정)

- **Notion**: 분류 결과 자동 동기화
- **Obsidian**: 마크다운 파일 연동
- **Google Drive**: 자동 백업

---

## 9. 📞 지원 및 문의

문제가 해결되지 않나요?


- **GitHub Issues**: [이슈 등록](#)
- **작성자**: [@jjaayy2222](#)
- **문서**: [README.md](#) 참조

---




## 10. 🔄 업데이트 내역

v3.5 (2025-11-11) - 현재 버전



- ☒ 스마트 온보딩 (GPT-4o 영역 추천)
- ☒ 맥락 기반 분류
- ☒ 실시간 대시보드
- ☒ 메타데이터 관리
- ☒ 검색 히스토리

-  마크다운 내보내기




### v3.0 (2025-11-01)

-  Dashboard 구현
-  SQLite 데이터베이스
-  MetadataAggregator

### v2.0 (2025-10-28)

-  PARA 분류 시스템
-  LangChain 통합

### v1.0 (2025-10-25)

-  기본 검색 기능
-  파일 업로드 자동화
-  FAISS 검색 엔진

---

**FlowNote**를 사용해 주셔서 감사합니다! 🎉

더 나은 경험을 위해 지속적으로 개선하고 있습니다.

피드백은 언제나 환영합니다! 💙

---

Made with  by [Jay](#)