

고블로EDMS 개발 계획 (개발환경 구축부터 단계별 진행)

1단계: 개발환경 구축 (1~2주)

✓ **목표:** 로컬 개발환경을 구축하고, 기본적인 시스템 아키텍처를 정리하여 협업이 가능하도록 한다.

작업 항목

1. 백엔드 환경 구축

- Docker Compose 설정 (PostgreSQL, Qdrant, Ollama, N8N)
- FastAPI 프로젝트 초기화 및 기본 API 테스트
- PostgreSQL 스키마 설계 (PARA 기반 문서 관리)
- API 문서화 (Swagger, Postman 설정)

2. 프론트엔드 환경 구축

- Flutter 프로젝트 생성 및 폴더 구조 정리
- 기본적인 화면 구성 (홈, 문서 목록, 편집기)
- REST API 연동 테스트
- Flutter 상태 관리 방식 결정 (Riverpod or Bloc)

3. CI/CD 및 코드 관리

- GitHub/GitLab 저장소 생성 및 코드 분리
 - Git Flow 전략 적용 (Main, Dev, Feature 브랜치 구조)
 - GitHub Actions 또는 GitLab CI/CD 설정
-

2단계: 백엔드 핵심 기능 개발 (3~4주)

✓ **목표:** 문서 업로드 및 관리 기능을 구현하고, AI 기반 자동 분류 및 검색을 가능하게 한다.

작업 항목

1. 문서 업로드 및 변환

- 파일 업로드 API 개발 (FastAPI)
- PDF, DOCX → Markdown 변환 모듈 개발
- AI 요약 기능 추가 (Ollama 연동)
- 문서 메타데이터 저장 (PostgreSQL)

2. PARA 시스템 적용

- 문서 자동 분류 알고리즘 개발
 - 문서 저장 시 PARA 카테고리 태깅
 - 사용자 지정 태그 기능 추가
3. 검색 기능 개발
- 키워드 검색 (PostgreSQL Full-Text Search)
 - 의미 기반 검색 (Qdrant)
 - OCR 기반 검색 추가 여부 검토 (Tesseract OCR 연동 가능)
4. 버전 관리 기능
- 문서 변경 이력 저장 (PostgreSQL)
 - 버전별 비교 기능 추가
 - 이전 버전 복구 기능 구현

3단계: 프론트엔드 주요 기능 개발 (4~5주)

목표: Markdown 기반 문서 작성 및 검색, 미리보기, 버전 관리를 지원하는 UI를 구현한다.

작업 항목

1. 문서 관리 대시보드 개발
 - 홈 화면: PARA 카테고리별 문서 정렬
 - 문서 목록 및 필터 UI 구현
 - 키워드 & AI 기반 검색 UI 추가
2. Markdown 문서 작성 및 편집
 - Markdown 편집기 적용 (flutter_markdown, zefyr 패키지 활용)
 - 실시간 미리보기 기능 구현
 - 파일 첨부 및 이미지 삽입 기능 추가
 - 태그 및 문서 메타데이터 추가 UI 구현
 - 자동 저장 기능 추가
3. 문서 보기 및 버전 관리
 - Markdown 문서 렌더링 기능
 - 첨부파일 미리보기 (PDF, 이미지 지원)
 - AI 요약 표시
 - 버전 비교 UI 구현 (Diff Highlighting)
 - 이전 버전 복원 기능 추가

4단계: 기능 고도화 및 테스트 (3~4주)

목표: 성능 및 보안을 최적화하고, 시스템 안정성을 높인다.

작업 항목

1. 성능 최적화
 - API 응답 속도 개선 (쿼리 최적화, 캐싱 적용)
 - Flutter UI 최적화 (로딩 속도 개선, 애니메이션 추가)
2. 통합 테스트
 - 백엔드 API 테스트 (Postman, PyTest 활용)
 - Flutter UI 테스트 진행
 - 파일 업로드 및 보안 점검
3. 검색 최적화
 - 검색 속도 개선 (인덱싱 최적화)
 - AI 기반 문서 추천 기능 추가

5단계: 배포 및 운영 (2주)

목표: 실환경에 배포하고, 사용자 피드백을 반영하며 지속적인 유지보수를 진행한다.

작업 항목

1. 클라우드 배포
 - AWS, GCP 또는 DigitalOcean 배포
 - Docker 기반 배포 환경 구축
 - 로드 밸런싱 및 Auto Scaling 설정
2. 운영 및 유지보수
 - 사용자 피드백 반영 및 기능 개선
 - 로그 분석 및 성능 모니터링 (Grafana, Prometheus)
 - 지속적인 보안 업데이트

17 **전체 일정 (4~5개월)**

단계	기간	주요 작업
1 개발환경 구축	1~2주	개발환경 설정, 컨테이너 실행, API 문서화
2 백엔드 기능 개발	3~4주	문서 업로드, Markdown 변환, AI 요약, 검색, 버전 관리
3 프론트엔드 개발	4~5주	문서 검색, 편집기, 미리보기, 버전 관리 UI
4 기능 고도화 및 테스트	3~4주	최적화, 보안 점검, 통합 테스트
5 배포 및 운영	2주	클라우드 배포, 유지보수, 모니터링

PARA 기반 EDMS 기대 효과

- ✓ 문서 관리가 목적에 따라 자동으로 정리됨 → 문서 찾기 쉬움
 - ✓ AI가 문서를 정리하고 요약해주므로 업무 생산성 향상
 - ✓ Markdown 기반으로 빠르고 효율적인 문서 작성 & 편집 가능
 - ✓ 버전 관리로 변경 이력 추적 및 복구 가능
 - ✓ 의미 기반 검색으로 필요한 문서를 빠르게 찾을 수 있음
-

결론

PARA 시스템을 적용한 고블로EDMS를 단계적으로 개발하면, 체계적인 문서 관리 및 검색이 가능한 AI 기반 EDMS를 완성할 수 있습니다!

 이제 1단계(개발환경 구성)부터 시작하면 될 것 같아요! 

어떤 부분부터 먼저 작업하는 것이 좋을까요? 😊