

# Legal GraphRAG

Local-first Privacy-focused Legal RAG

검증 가능한 법률 AI 어시스턴트

# 1. 프로젝트 개요

## Vision

"Legal GraphRAG (Local-first)"

법률 지식 그래프(KG)와 벡터 검색을 결합하여, **검증 가능하고 근거가 확실한 법**  
률 답변을 제공하는 로컬 우선 RAG 시스템

## 핵심 가치

-  **Local-First:** 데이터 유출 걱정 없는 로컬 환경 (Ollama, Neo4j, Chroma)

# 2. 핵심 목표

## 1. 로컬 보안 (Privacy)

- 모든 인프라를 로컬 머신에서 구동
- 민감한 법률/내부 문서 보호

## 2. 정확성 (Accuracy)

- 하이브리드 검색으로 "환각(Hallucination)" 최소화
- 그래프 탐색으로 법적 연결성(준용/참조) 고려

## 3. 검증 가능성 (Groundedness)

- 사용자가 원문을 바로 확인할 수 있도록 **인용** 필수 제공

# 3. 기술 스택 (Local Stack)

구분	기술	설명
Lang	Python 3.10+	주 개발 언어
G-DB	<a href="#">Neo4j Desktop</a>	지식 그래프, 시각화
V-DB	<a href="#">Chroma</a>	경량 벡터 저장소
LLM	<a href="#">Ollama</a>	Llama 3, Qwen 2.5 (Local)
UI	Streamlit	빠른 프로토타이핑

# 4. 시스템 아키텍처



Syntax error in text  
mermaid version 11.12.2

API 수집 → 정규화 → 그래프(Neo4j)/벡터(Chroma) 적재 → 하이브리드 검색 →  
*LLM* 생성

# 5. 데이터 파이프라인

1. **Ingestion**: 법제처 API 수집 (JSON)
2. **Normalization**: Law / Article 분리 및 정규화
3. **Construction**: Neo4j 노드/엣지 생성
  - (Law)-[ :HAS ]->(Article)
  - (Case)-[ :CITES ]->(Article)
4. **Embedding**: 판례/문서 Chunking & Vectorization

# 6. 하이브리드 검색 전략

"Graph로 범위를 좁히고, Vector로 정밀도를 높인다"

## 1. Step A: Graph Retrieval

- 검색어 ("해지 환급금") → 엔티티 추출 → 그래프 2-hop 탐색
- 관련 조문/개념 후보군 확보 (High Recall)

## 2. Step B: Vector Scoring

- 그래프 후보 + 벡터 검색 후보 결합
- 유사도 기반 재순위화 (Reranking)

# 7. 데이터 모델 (Schema)

- **Nodes**

- Law (법령)
- Article (조문)
- Case (판례)
- Concept (개념 - *Phase 2*)
- DocumentChunk (벡터 청크)

- **Edges**

- HAS\_ARTICLE , REFERS\_TO , CITES , DEFINED\_IN , MENTIONS

# 8. 성공 지표 (Metrics)

- 검색 성능

-  **Hit@K**: 정답 조문이 Top-5 안에 있는가?
-  **Graph Coverage**: 그래프 연결성 활용도

- 답변 품질

-  **Citation Rate**: 인용이 포함된 문장 비율
-  **Groundedness**: 답변의 근거가 검색 결과에 있는가?

# 9. 로드맵

## 1. Phase 1 (MVP) - *Current*

- 기본 파이프라인 구축 & 하이브리드 검색 구현
- Streamlit 데모 완성

## 2. Phase 2 (Extension)

- 개념(Concept) 그래프 확장
- 심화 평가 (Ragas) 및 UI 고도화

# 감사합니다

질문 있으신가요?

GitHub: <https://github.com/hyeonz673/legal-graph-rag>