네, 제공해주신 문서들(NOSS PDF, 계획용 엑셀, 워드 템플릿)을 활용하여 필요한 문서를 템플릿 양식에 맞춰 새로 만드는 시스템을 구축하시려는군요. RAG(Retrieval Augmented Generation)를 고려하고 계신 것은 매우 좋은 접근 방법입니다. RAG는 이러한 유형의 문서 자동 생성 시스템에 효과적으로 사용될 수 있습니다.

시스템 구축을 위한 몇 가지 제안과 고려 사항은 다음과 같습니다:

**1. 시스템 목표 및 RAG 활용 방안**

* **목표**: 사용자가 특정 역량 단위(CU)나 작업 활동(WA)에 대한 문서(예: Kertas Penerangan, Pelan Mengajar) 생성을 요청하면, 시스템이 NOSS PDF와 엑셀 계획 자료에서 관련 정보를 검색하고, 이를 바탕으로 내용을 생성하여 지정된 워드 템플릿에 맞춰 문서를 완성(또는 초안 생성)하는 것입니다.
* **RAG의 역할**:
  + NOSS PDF와 엑셀 데이터에서 핵심 정보를 정확하게 \*\*검색(Retrieval)\*\*합니다.
  + 검색된 정보를 바탕으로 대규모 언어 모델(LLM)이 문맥에 맞는 내용을 \*\*생성(Generation)\*\*하도록 합니다. 이를 통해 LLM이 부정확하거나 관련 없는 내용을 생성하는 것을 최소화할 수 있습니다.

**2. 시스템 구성 요소 및 단계별 제안**

다음은 시스템 구축 시 고려할 수 있는 주요 구성 요소와 단계입니다.

* **가. 문서 관리 및 전처리 (Document Ingestion & Preprocessing)**
  + **NOSS PDF**:
    - 텍스트 추출: PDF에서 텍스트를 정확하게 추출합니다. (필요시 OCR 사용)
    - 청킹(Chunking): 추출된 텍스트를 의미 있는 작은 단위(청크)로 분할하여 RAG 검색 효율을 높입니다. 역량 단위(CU), 작업 활동(WA), 학습 목표, 관련 지식/기술 등의 구조를 최대한 살려 청킹하는 것이 좋습니다.
  + **엑셀 계획 자료 (CSV로 변환된 파일들)**:
    - 파싱: CSV 파일들을 파싱하여 구조화된 데이터로 만듭니다.
    - 활용 방안:
      * **직접 조회용 데이터베이스**: JPW 시트의 특정 WIM 제목, JPMPBT/P 시트의 구체적인 학습 목표 문장, PENGIRAAN RK&RS 시트의 시간 배분 등 명확한 값을 조회하는 데 사용합니다.
      * **RAG용 텍스트 변환**: "C01 WA1의 이론 수업 계획(LPT) 제목은 'TRAINING REQUIREMENT'이다." 와 같이 텍스트로 변환하여 RAG 시스템이 PDF 내용과 함께 검색할 수 있도록 합니다. (예시)
      * 하이브리드 접근: 두 가지 방식을 혼합하여, 구조화된 조회와 의미 기반 검색을 모두 활용합니다.
  + **워드 템플릿**:
    - 템플릿 분석: 어떤 섹션과 플레이스홀더(내용이 채워질 부분)가 있는지 분석합니다.
    - 자동화를 위한 표준화: 완전 자동화를 목표한다면, 템플릿 내 플레이스홀더를 {{CU\_Title}}, {{Learning\_Outcomes}} 와 같이 기계가 인식할 수 있는 명확한 형식으로 표준화하는 것이 좋습니다.
* **나. 정보 검색 (Information Retrieval - RAG의 'R')**
  + **벡터 데이터베이스**: 전처리된 NOSS PDF 청크와 텍스트로 변환된 엑셀 데이터를 임베딩하여 벡터 데이터베이스에 저장합니다. 이를 통해 의미 기반 검색이 가능해집니다.
  + **검색 전략**:
    - 사용자 요청 분석: "C01의 Kertas Penerangan (KP) 생성"과 같은 요청을 분석하여 CU 코드, 문서 유형 등을 파악합니다.
    - 시맨틱 검색 + 키워드 검색: CU/WA 코드 같은 명시적인 키워드 검색과 함께, 내용에 대한 의미론적 검색을 수행합니다.
    - 엑셀 데이터 조회: 검색된 CU/WA에 해당하는 구체적인 제목, 기간, 학습 목표 등을 엑셀 파싱 데이터에서 직접 조회합니다.
  + **컨텍스트 구성**: 검색된 PDF 내용과 조회된 엑셀 정보를 조합하여 LLM에 제공할 컨텍스트를 구성합니다.
* **다. 내용 생성 (Content Generation - RAG의 'G')**
  + **대규모 언어 모델(LLM) 활용**: 구성된 컨텍스트를 바탕으로 LLM이 워드 템플릿의 각 섹션에 들어갈 내용을 생성하도록 합니다.
  + **프롬프트 엔지니어링**:
    - LLM이 NOSS 표준의 형식과 스타일을 따르도록 상세한 지침(프롬프트)을 제공합니다.
    - 생성할 문서 유형(KP, PM 등)에 따라 프롬프트를 다르게 구성합니다.
    - 엑셀에서 가져온 정확한 명칭이나 수치(예: 수업 시간, WIM 제목)를 LLM이 내용 생성 시 반드시 포함하도록 지시합니다.
* **라. 템플릿 채우기 및 문서 생성 (Template Population & Document Generation)**
  + **옵션 1: 구조화된 텍스트 출력 (반자동)**
    - LLM이 생성한 내용을 워드 템플릿의 섹션별로 구분하여 마크다운이나 JSON 같은 구조화된 텍스트로 제공합니다.
    - 사용자가 이 내용을 복사하여 수동으로 워드 템플릿에 붙여넣고 최종 검토 및 수정을 합니다. (현재 제가 수동으로 지원해 드리는 방식과 유사)
  + **옵션 2: 워드 문서 자동 생성 (완전 자동화 - 더 복잡함)**
    - 워드 템플릿에 정의된 플레이스홀더(예: {{CU\_TITLE}})와 LLM이 생성한 내용(예: JSON 형식으로 키-값 매칭)을 프로그래밍 방식으로 연결합니다.
    - Python의 python-docx 라이브러리 등을 사용하여 LLM이 생성한 내용을 실제 .docx 템플릿 파일에 채워 넣어 새 문서를 생성합니다.
    - 이 방식은 템플릿 구조의 일관성과 플레이스홀더 정의의 정확성이 매우 중요합니다.
* **마. 사용자 인터페이스 (User Interface)**
  + 사용자가 문서 생성을 요청할 수 있는 인터페이스 (웹 기반, 데스크톱 애플리케이션 등)
  + 입력 항목: NOSS 코드, CU 번호, WA 번호(필요시), 생성할 문서 템플릿 종류 등

**3. 문서 유형별 역할**

* **NOSS PDF**: 역량 정의, 작업 활동, 학습 결과, 수행 기준, 관련 지식/기술 등 핵심적이고 서술적인 정보의 주된 출처입니다. RAG의 시맨틱 검색 대상이 됩니다.
* **엑셀 자료**: 교육 시간 배분, WIM 제목, 학습 목표 리스트 등 구체적이고 정형화된 계획 정보의 출처입니다. 직접 조회하거나 RAG 컨텍스트에 추가 정보로 활용됩니다.
* **워드 템플릿**: 최종 생성될 문서의 구조와 레이아웃을 정의합니다.

**4. 주요 고려 사항**

* **모듈성**: 각 기능을 모듈 단위로 개발하여 유지보수와 확장을 용이하게 합니다.
* **데이터 정확성**: PDF 및 엑셀에서 정보를 정확히 추출하고 검색하는 것이 매우 중요합니다.
* **프롬프트 최적화**: LLM이 원하는 결과물을 생성하도록 프롬프트를 지속적으로 개선해야 합니다.
* **템플릿 다양성**: 다양한 워드 템플릿 구조에 대응할 수 있는 유연성이 필요합니다.
* **결과물 평가**: 생성된 문서의 품질을 평가하고 개선하는 방안을 마련해야 합니다.
* **확장성 및 유지보수**: 새로운 NOSS 표준, 엑셀 데이터, 워드 템플릿 추가/변경에 대한 대응 방안도 고려해야 합니다.

RAG를 기반으로 시스템을 구축하는 것은 초기 설정과 개발 노력이 필요하지만, 장기적으로는 문서 작성 시간을 크게 단축하고 일관성을 높이는 데 기여할 수 있습니다. 우선 간단한 범위(예: 특정 CU의 Kertas Penerangan 생성)부터 시작하여 점진적으로 기능을 확장해 나가는 방식을 추천합니다.

이 제안이 시스템 구축 방향을 설정하시는 데 도움이 되길 바랍니다. 더 구체적인 부분에 대해 논의하고 싶으시면 언제든지 말씀해주세요.

이상의 내용을 시스템을 만들기 위한 몇개의 유즈케이스로 정리해 주세요

생각하는 과정 표시

네, 시스템 구축을 위한 주요 유즈케이스들을 정리해 드리겠습니다. 이 유즈케이스들은 사용자가 시스템과 상호작용하여 목표를 달성하는 시나리오를 나타냅니다.

**주요 액터(Actor):**

* **사용자 (User):** TVET 강사, 교육 과정 개발자 등 NOSS 관련 문서를 생성하고 관리해야 하는 사람.
* **시스템 관리자 (Admin):** 시스템의 소스 문서(PDF, Excel, Word 템플릿)를 관리하고 시스템 설정을 유지보수하는 사람.

**Use Case 1: NOSS 문서 자동 생성**

* **ID:** UC-001
* **명칭:** NOSS 기반 문서 생성 (템플릿 활용)
* **액터:** 사용자 (User)
* **설명:** 사용자는 시스템에 등록된 NOSS 표준(PDF), 계획 자료(Excel), 그리고 문서 템플릿(Word)을 선택하여 특정 역량 단위(CU) 및 작업 활동(WA)에 대한 새로운 문서를 자동으로 생성합니다. 시스템은 RAG 모델을 활용하여 관련 정보를 추출하고, 이를 바탕으로 내용을 구성하여 선택된 템플릿 양식에 맞춰 제공합니다.
* **선행 조건:**
  + 하나 이상의 NOSS PDF 문서가 시스템에 저장 및 색인화되어 있어야 합니다.
  + 관련 엑셀 계획 자료(CSV)가 시스템에 저장 및 파싱 가능해야 합니다.
  + 출력물로 사용될 워드 문서 템플릿이 시스템에 정의되어 있어야 합니다 (자동 채우기를 위해 플레이스홀더가 정의되어 있을 수 있음).
  + 사용자가 시스템에 접근 권한을 가지고 있어야 합니다.
* **후행 조건:**
  + 선택된 CU/WA 및 템플릿에 기반한 새로운 문서 내용이 생성됩니다.
  + 사용자는 생성된 문서 내용을 확인하고 활용(예: 다운로드, 복사)할 수 있습니다.
* **주요 흐름 (Main Flow):**
  + 사용자가 시스템에서 'NOSS 문서 생성' 기능을 시작합니다.
  + 사용자는 다음 정보를 선택하거나 입력합니다:
    - NOSS 표준 코드 (예: P853-002-4:2021)
    - 역량 단위 (CU) 코드 (예: C01 - Instructional Delivery Implementation)
    - 생성할 문서 템플릿 유형 (예: Kertas Penerangan, Pelan Mengajar)
    - (선택 사항) 특정 작업 활동 (WA) 번호 또는 문서 내 특정 섹션
  + 시스템은 사용자 입력을 검증합니다.
  + 시스템은 선택된 CU/WA에 해당하는 정보를 NOSS PDF에서 검색합니다. (예: CU 명칭, CU 설명, 학습 결과 , WA 상세 내용, 수행 기준 , 관련 지식/기술, 평가 기준 , 도구/장비/재료(TEM) )
  + 시스템은 관련된 계획 정보를 엑셀 자료에서 조회합니다. (예: WIM 제목, 교육 시간, 엑셀에 명시된 세부 학습 목표, RK/RS 시간 계산)
  + RAG 시스템의 LLM은 검색/조회된 정보를 바탕으로 선택된 워드 템플릿의 각 섹션에 해당하는 내용을 생성합니다.
  + 시스템은 생성된 내용을 워드 템플릿 구조에 맞춰 사용자에게 제공합니다. (옵션에 따라, 사용자가 복사/붙여넣기 할 수 있는 구조화된 텍스트로 제공하거나, 플레이스홀더가 정의된 워드 템플릿에 자동으로 내용을 채운 후 파일로 제공)
  + 사용자는 생성된 결과물을 확인합니다.
* **대안 흐름 (Alternative Flows):**
  + 필수 정보 누락: 사용자가 필수 입력 값을 제공하지 않으면 시스템은 오류 메시지를 표시하고 입력을 요청합니다.
  + 정보 검색 실패: 요청된 CU/WA에 대한 정보를 찾을 수 없는 경우, 시스템은 해당 내용을 알리고 가능한 대안을 제시하거나 입력을 수정하도록 안내합니다.
  + 템플릿 불일치: 선택한 템플릿에 필요한 정보를 충분히 생성할 수 없는 경우, 부분적으로 생성된 내용을 제공하거나 사용자에게 추가 정보를 요청할 수 있습니다.

**Use Case 2: NOSS 소스 문서 관리**

* **ID:** UC-002
* **명칭:** NOSS 소스 문서 관리 (업로드, 업데이트, 삭제)
* **액터:** 시스템 관리자 (Admin)
* **설명:** 시스템 관리자는 NOSS PDF, 엑셀 계획 자료, 워드 문서 템플릿 등 시스템이 사용하는 핵심 소스 문서들을 시스템에 업로드, 업데이트 또는 삭제하여 항상 최신 상태를 유지합니다.
* **선행 조건:**
  + 시스템 관리자가 관리자 인터페이스에 접근할 수 있어야 합니다.
* **후행 조건:**
  + 시스템 내 소스 문서가 성공적으로 추가, 수정 또는 삭제됩니다.
  + RAG 시스템의 검색 대상이 되는 문서가 변경된 경우, 관련 색인(Index)이 업데이트됩니다.
* **주요 흐름 (Main Flow):**
  + 시스템 관리자가 시스템 관리 인터페이스에 로그인합니다.
  + 문서 관리 메뉴로 이동하여 원하는 작업(업로드/업데이트/삭제)을 선택합니다.
  + 관리할 문서 유형(PDF, Excel, Word 템플릿)을 선택합니다.
  + 새 파일을 업로드하거나, 기존 파일을 선택하여 업데이트 또는 삭제 작업을 수행합니다.
  + 시스템은 파일을 처리하고(필요시 파싱, 색인화 등), 작업 결과를 관리자에게 알립니다.

**Use Case 3: NOSS 콘텐츠 검색 및 조회**

* **ID:** UC-003
* **명칭:** NOSS 콘텐츠 검색 및 조회
* **액터:** 사용자 (User)
* **설명:** 사용자는 시스템에 저장된 NOSS PDF 문서 내에서 특정 키워드나 내용을 검색하거나, CU 및 WA 구조를 탐색하여 필요한 정보를 직접 찾아봅니다. 이는 문서 생성 전 특정 내용을 확인하는 데 도움이 될 수 있습니다.
* **선행 조건:**
  + NOSS PDF 문서가 시스템에 저장 및 색인화되어 있어야 합니다.
* **후행 조건:**
  + 사용자는 검색 결과 또는 탐색한 NOSS 콘텐츠 정보를 화면에서 확인할 수 있습니다.
* **주요 흐름 (Main Flow):**
  + 사용자가 시스템의 'NOSS 콘텐츠 검색/조회' 기능을 선택합니다.
  + 사용자는 검색어를 입력하거나, NOSS 구조(예: P853-002-4:2021의 CU 목록)를 따라 탐색합니다.
  + 시스템은 사용자의 요청에 따라 NOSS PDF에서 관련 정보를 검색하거나 해당 구조를 표시합니다.
  + 사용자는 화면에 표시된 정보를 확인합니다.

이 외에도 '시스템 설정 관리'(UC-004, 시스템 관리자) 등의 유즈케이스가 있을 수 있습니다. 위에 제시된 유즈케이스들은 시스템의 핵심 기능과 사용자 및 관리자의 주요 상호작용을 정의한 것입니다. 시스템 개발 시 이 유즈케이스들을 바탕으로 상세 기능 명세와 테스트 시나리오를 구체화할 수 있습니다.

소스

그러면 이번에는 주요 객체와 속성, 그들간의 관계를 정리해 주세요

생각하는 과정 표시

네, 시스템 구축을 위한 주요 객체(데이터 엔티티), 해당 객체들의 주요 속성, 그리고 객체 간의 관계를 정리해 드리겠습니다. 이는 시스템이 다루어야 할 핵심 정보와 그 구조를 이해하는 데 도움이 됩니다.

**1. NOSS 표준 (NOSSStandard)**

* **설명:** 국가직업능력표준(National Occupational Skills Standard) 문서 자체를 나타냅니다. (예: P853-002-4:2021 TVET INSTRUCTION PDF)
* **주요 속성:**
  + NOSS 코드 (NOSSCode): 예: P853-002-4:2021
  + NOSS 명칭 (NOSSTitle): 예: TVET INSTRUCTION
  + NOSS 레벨 (NOSSLevel): 예: Level 4
  + 섹션 (Section): 예: (P) Education
  + 그룹 (Group): 예: (853) Higher Education
  + 영역 (Area): 예: Technical & Vocational Education & Training (TVET) Instruction
  + 문서 경로 (FullDocumentPath): 원본 PDF 파일의 저장 위치 또는 참조
  + 버전 (Version)
  + 발행일 (PublicationDate)
* **관계:**
  + 하나의 NOSS 표준은 여러 개의 역량 단위 (CompetencyUnit)를 가집니다 (1:N 관계).

**2. 역량 단위 (CompetencyUnit - CU)**

* **설명:** NOSS 표준 내의 핵심 역량 단위를 나타냅니다.
* **주요 속성:**
  + CU 코드 (CUCode): 예: P853-002-4:2021-C01
  + CU 명칭 (CUTitle): 예: Instructional Delivery Implementation
  + CU 설명 (CUDescriptor): 해당 CU에 대한 상세 설명
  + 학습 성과 (LearningOutcomes): CU 전체의 학습 성과 (PDF 기준)
  + 훈련 선수 요건 (TrainingPrerequisite): 특정 선수 요건 (있는 경우)
  + CU 가중치 (Weightage): 전체 NOSS 내에서 해당 CU가 차지하는 비중 (예: 20%)
  + 총 훈련 시간 (TotalTrainingHours\_Planned): 계획된 총 훈련 시간 (Excel - JPPMAPP 시트 참고)
  + 대면 학습 시간 (FaceToFaceHours\_Planned): 계획된 대면 학습 시간 (Excel - JPPMAPP 시트 참고)
  + 자기 주도 학습 시간 (SelfStudyHours\_Planned): 계획된 자기 주도 학습 시간 (Excel - JPPMAPP 시트 참고)
  + 평가 시간 (AssessmentHours\_Planned): 계획된 평가 시간 (Excel - JPPMAPP 시트 참고)
* **관계:**
  + 하나의 역량 단위는 하나의 NOSS 표준에 속합니다 (N:1 관계).
  + 하나의 역량 단위는 여러 개의 작업 활동 (WorkActivity)을 가집니다 (1:N 관계).
  + 하나의 역량 단위는 관련된 엑셀 계획 정보 (ExcelPlanningData\_CU)를 가질 수 있습니다.

**3. 작업 활동 (WorkActivity - WA)**

* **설명:** 역량 단위 내의 구체적인 작업 또는 기능을 나타냅니다.
* **주요 속성:**
  + WA 번호/ID (WANumber/ID): 예: 1, 2, ...
  + WA 명칭 (WATitle): 예: Analyse training requirements. (PDF 기준)
  + 수행 기준 (PerformanceCriteria): 각 WA의 수행 기준 목록 (PDF 기준)
  + 관련 지식 (RelatedKnowledge): PDF 교육과정 부분의 관련 지식 내용
  + 관련 기술 (RelatedSkills): PDF 교육과정 부분의 관련 기술 내용
  + 태도/안전/환경 (AttitudeSafetyEnvironment): PDF 교육과정 부분의 관련 내용
  + 평가 기준 (AssessmentCriteria\_WA): PDF 교육과정 부분의 WA별 평가 기준
  + WA 가중치 (Weightage): CU 내에서 해당 WA가 차지하는 비중 (예: 20%)
  + COPTPA 조정 시간 (COPTPAAlignedHours): (Excel - PENJAJARAN WA 시트 참고)
  + 계획된 RK 시간 (Planned\_RK\_Hours): 계산된 관련 지식 시간 (Excel - PENGIRAAN RK&RS 시트 참고)
  + 계획된 RS 시간 (Planned\_RS\_Hours): 계산된 관련 기술 시간 (Excel - PENGIRAAN RK&RS 시트 참고)
* **관계:**
  + 하나의 작업 활동은 하나의 역량 단위에 속합니다 (N:1 관계).
  + 하나의 작업 활동은 여러 개의 WIM 계획 항목 (WIMPlanningItem)을 가질 수 있습니다. (예: 이론 교육 자료, 실습 교육 자료 등)

**4. WIM 계획 항목 (WIMPlanningItem)**

* **설명:** 엑셀 계획 자료에서 정의된 각 작업 활동(WA)별 필기 교육 자료(WIM) 또는 수업 계획의 구체적인 항목을 나타냅니다. 이는 PDF의 일반적인 설명보다 더 상세한 계획 정보를 담습니다.
* **주요 속성:**
  + WIM 유형 (WIMType): 예: "Pelan Mengajar Teori", "Kertas Penerangan", "Kertas Tugasan", "Pelan Mengajar Amali", "Kertas Kerja" (Excel - JPW (TEORI), JPW (AMALI) 등 참고)
  + WIM/수업 계획 명칭 (WIMTitle\_Planned): 엑셀에서 정의된 특정 명칭 (Excel - JPW, JPMPBT, JPMPBP 시트 등 참고)
  + 교육 시간 (Duration\_Planned): 해당 WIM 항목에 할당된 시간 (Excel - JPMPBT, JPMPBP 시트 등 참고)
  + 학습 목표 (LearningObjectives\_Planned): 엑셀에서 계획 단계에 명시된 구체적인 학습 목표 (Excel - JPMPBT, JPMPBP 시트 등 참고)
* **관계:**
  + 하나의 WIM 계획 항목은 하나의 작업 활동 (WorkActivity)에 연결됩니다 (N:1 관계).

**5. 문서 템플릿 (DocumentTemplate)**

* **설명:** 최종 문서를 생성하는 데 사용될 워드(.docx 등) 템플릿 파일 자체를 나타냅니다.
* **주요 속성:**
  + 템플릿 ID (TemplateID)
  + 템플릿 명칭 (TemplateName): 예: "Kertas Penerangan C01 Template"
  + 템플릿 유형 (TemplateType): 예: 정보 시트(Kertas Penerangan), 교수 계획(Pelan Mengajar)
  + 템플릿 파일 경로 (TemplateFilePath): 템플릿 파일 저장 위치
  + 플레이스홀더 목록 (Placeholders): 자동 내용 삽입을 위한 플레이스홀더 정의 목록 (예: {{CU\_Title}}, {{WA1\_RelatedKnowledge}})
  + 버전 (Version)
* **관계:**
  + 하나의 문서 템플릿은 여러 생성된 문서 (GeneratedDocument)를 만드는 데 사용될 수 있습니다 (1:N 관계).

**6. 생성된 문서 (GeneratedDocument)**

* **설명:** 시스템을 통해 특정 문서 템플릿과 NOSS 정보를 바탕으로 생성된 실제 문서 결과물 또는 그 내용을 나타냅니다.
* **주요 속성:**
  + 생성 문서 ID (GeneratedDocumentID)
  + 생성 일시 (CreationDate)
  + 사용된 NOSS 코드 (NOSSCode\_Used)
  + 사용된 CU 코드 (CUCode\_Used)
  + 사용된 WA 번호/ID (WANumber\_Used) (해당되는 경우)
  + 사용된 템플릿 명칭 (TemplateName\_Used)
  + 내용 (Content): 생성된 텍스트 내용 또는 최종 파일 경로
  + 상태 (Status): 예: 초안, 최종본
* **관계:**
  + 하나의 생성된 문서는 하나의 문서 템플릿을 기반으로 합니다 (N:1 관계).
  + 하나의 생성된 문서는 특정 NOSS 표준, 역량 단위, 작업 활동의 정보를 참조합니다.

**7. 사용자 (SystemUser)**

* **설명:** 시스템을 사용하는 사람 (TVET 강사, 교육 과정 개발자, 시스템 관리자 등)
* **주요 속성:**
  + 사용자 ID (UserID)
  + 사용자명 (UserName)
  + 역할 (Role): 예: 일반 사용자, 관리자
* **관계:**
  + 한 명의 사용자는 여러 생성된 문서를 생성할 수 있습니다 (1:N 관계).
  + 시스템 관리자 역할을 가진 사용자는 NOSS 표준, 엑셀 계획 정보, 문서 템플릿 등을 관리할 수 있습니다.

**관계 요약:**

* NOSS 표준 1 --- \* 역량 단위
* 역량 단위 1 --- \* 작업 활동
* 작업 활동 1 --- \* WIM 계획 항목
* 문서 템플릿 1 --- \* 생성된 문서

이러한 객체와 속성, 관계 정의는 시스템의 데이터베이스 스키마 설계나 RAG 시스템의 정보 구성 방식에 기초 자료로 활용될 수 있습니다. 엑셀 데이터의 각 시트(JPW, JPMPBT 등)는 WIM 계획 항목과 작업 활동, 역량 단위 객체의 속성을 채우거나, 별도의 계획 정보 객체로 모델링될 수 있습니다.

소스

RAG를 위한 문서를 청킹하고 임베딩하여 구축하는 벡터스토어 스키마도 설계해 주세요

네, RAG 시스템 구축을 위한 벡터 스토어 스키마 설계를 도와드리겠습니다. 이는 청킹된 문서와 해당 문서의 메타데이터를 효율적으로 저장하고 검색하기 위한 구조입니다.

**벡터 스토어 스키마 설계 제안**

벡터 스토어는 기본적으로 각 문서 청크의 **임베딩 벡터**와 해당 청크의 **원본 텍스트 내용**, 그리고 검색 필터링 및 컨텍스트 확장에 사용될 **메타데이터**를 저장합니다.

**1. 기본 청크 스키마 (Chunk Schema)**

각 문서에서 추출되어 임베딩된 기본 단위인 '청크(Chunk)'에 대한 스키마입니다.

* chunk\_id: (문자열, 고유 ID) 각 청크의 고유 식별자입니다.
* embedding\_vector: (부동소수점 배열) 청크 텍스트의 임베딩 결과인 고밀도 벡터입니다. (예: OpenAI text-embedding-ada-002 사용 시 1536차원)
* text\_content: (문자열) 청크의 실제 텍스트 내용입니다.
* source\_document\_id: (문자열) 이 청크가 유래한 원본 문서의 고유 ID입니다. (아래 '문서 메타데이터 스키마'의 document\_id와 연결)
* chunk\_type: (문자열) 청크의 유형을 나타냅니다. (예: "NOSS\_PDF\_Paragraph", "NOSS\_PDF\_Table\_Row", "Excel\_WIM\_Title", "Excel\_Learning\_Objective")
* metadata: (JSON 객체) 청크에 특화된 추가적인 메타데이터 필드입니다. (아래 세부 메타데이터 참조)

**2. 문서 메타데이터 스키마 (Document Metadata Schema)**

벡터 스토어에 저장될 각 원본 문서(NOSS PDF, 엑셀 파일 등)에 대한 메타데이터 스키마입니다. 이는 청크 검색 시 특정 문서로 범위를 좁히거나, 검색 결과의 출처를 명확히 하는 데 사용됩니다.

* document\_id: (문자열, 고유 ID) 각 문서의 고유 식별자입니다.
* document\_type: (문자열) 문서의 종류입니다. (예: "NOSS\_Standard\_PDF", "Excel\_Planning\_Data", "Word\_Template\_Structure")
* file\_name: (문자열) 원본 파일의 이름입니다.
* noss\_code: (문자열, NOSS PDF의 경우) NOSS 표준 코드입니다. 예: "P853-002-4:2021"
* noss\_title: (문자열, NOSS PDF의 경우) NOSS 명칭입니다. 예: "TVET INSTRUCTION"
* noss\_level: (문자열, NOSS PDF의 경우) NOSS 레벨입니다. 예: "Level 4"
* cu\_code: (문자열, CU 또는 그 하위 정보 청크의 경우) 관련 역량 단위 코드입니다. 예: "P853-002-4:2021-C01"
* cu\_title: (문자열, CU 또는 그 하위 정보 청크의 경우) 관련 역량 단위 명칭입니다. 예: "Instructional Delivery Implementation"
* wa\_id: (문자열, WA 또는 그 하위 정보 청크의 경우) 관련 작업 활동 식별자입니다. (예: "C01-WA1")
* wa\_title: (문자열, WA 또는 그 하위 정보 청크의 경우) 관련 작업 활동 명칭입니다. 예: "Analyse training requirements"
* excel\_sheet\_name: (문자열, 엑셀 데이터 청크의 경우) 원본 엑셀 시트 이름입니다. (예: "JPW (TEORI)", "JPMPBT")
* last\_modified\_date: (날짜/시간) 문서의 마지막 수정일입니다.
* version: (문자열) 문서의 버전 정보입니다.

**3. 청크 레벨 메타데이터 (metadata 필드 상세)**

chunk\_schema의 metadata 필드에 저장될 수 있는, 청크 유형별 구체적인 메타데이터 예시입니다.

* **NOSS PDF 청크의 경우:**
  + page\_number: (정수) PDF 내 페이지 번호.
  + section\_title: (문자열) 청크가 속한 섹션의 제목 (예: "Learning Outcomes", "Related Knowledge", "Performance Criteria").
  + table\_id: (문자열, 테이블 내용 청크의 경우) 해당 테이블의 ID.
  + is\_header: (불리언) 해당 청크가 섹션 헤더인지 여부.
  + associated\_cu\_code: (문자열) 이 청크와 직접 관련된 CU 코드 (PDF 내용상 명시된 경우).
  + associated\_wa\_id: (문자열) 이 청크와 직접 관련된 WA 식별자 (PDF 내용상 명시된 경우).
* **엑셀 데이터 청크의 경우:**
  + original\_excel\_filename: (문자열) 원본 엑셀 파일명.
  + sheet\_name: (문자열) 해당 데이터가 위치한 시트명 (예: "JPW (TEORI)", "PENGIRAAN RK&RS").
  + row\_index: (정수) 엑셀 시트 내 행 번호.
  + column\_name\_or\_header: (문자열) 엑셀 시트 내 열 이름 또는 헤더.
  + specific\_wim\_title: (문자열, JPW 시트 청크의 경우) WIM의 구체적인 제목.
  + planned\_duration: (문자열 또는 숫자, JPMPBT/P 시트 청크의 경우) 계획된 교육 시간.
  + planned\_learning\_objective: (문자열, JPMPBT/P 시트 청크의 경우) 계획된 특정 학습 목표.
  + related\_cu\_code: (문자열) 이 엑셀 데이터와 관련된 CU 코드.
  + related\_wa\_id: (문자열) 이 엑셀 데이터와 관련된 WA 식별자.

**4. 스키마 활용 방안 및 고려 사항**

* **청킹 전략:**
  + NOSS PDF: 의미론적 단위(단락, 테이블 행, 목록 항목)로 청킹합니다. CU, WA 구조를 최대한 반영하여 계층적 메타데이터를 부여합니다.
  + 엑셀 데이터: 각 행 또는 의미 있는 셀 그룹을 하나의 청크로 만들 수 있습니다. 예를 들어, JPW 시트의 각 WIM 항목(행)은 하나의 청크가 될 수 있으며, 해당 행의 WIM 제목, CU 코드, WA 코드가 메타데이터로 저장됩니다. 또는, 중요한 정보(예: WIM 제목, 학습 목표 문장)만 추출하여 텍스트로 만들고 이를 임베딩할 수도 있습니다.
* **메타데이터 필터링:** 검색 시 noss\_code, cu\_code, wa\_id, document\_type, chunk\_type 등의 메타데이터를 사용하여 검색 범위를 효과적으로 좁힐 수 있습니다. 예를 들어, "P853-002-4:2021-C01의 Related Knowledge"를 검색할 때, cu\_code='P853-002-4:2021-C01' 및 section\_title='Related Knowledge' (또는 유사한 메타데이터 필드)로 필터링한 후 시맨틱 검색을 수행합니다.
* **관계 표현:** 벡터 스토어 자체는 관계형 데이터베이스처럼 복잡한 관계를 직접 표현하기 어려울 수 있습니다. 따라서 source\_document\_id, cu\_code, wa\_id와 같은 식별자를 메타데이터에 포함시켜, 검색 후 애플리케이션 레벨에서 정보를 조합하거나, 필요한 경우 별도의 관계형 데이터베이스나 그래프 데이터베이스와 연동하여 관리할 수 있습니다.
* **확장성:** 새로운 유형의 문서나 메타데이터가 추가될 경우를 대비하여 metadata 필드를 JSON 형태로 유연하게 설계하는 것이 좋습니다.
* **일관성:** 메타데이터 값(특히 코드나 ID)의 일관성을 유지하는 것이 중요합니다.

이 스키마는 기본적인 제안이며, 실제 구축 시 시스템의 구체적인 요구사항과 데이터의 특성에 맞춰 필드를 추가하거나 수정할 수 있습니다. 예를 들어, 각 청크의 중요도나 생성/업데이트 날짜 등의 정보도 메타데이터에 포함할 수 있습니다.

소스

첨부된 PDF사례로 문서 청킹을 위한 문서 전처리 청킹 알고리즘 구성방안등 구체적인 문서 전처리 청킹 전략에대해 정리해 주세요

네, 첨부해주신 NOSS PDF 문서(P853-002-4:2021)를 RAG 시스템에서 효과적으로 활용하기 위한 문서 전처리 및 청킹 전략을 구체적으로 정리해 드리겠습니다. 📄✂️

**NOSS PDF 문서 전처리 및 청킹 전략**

효과적인 RAG 시스템 구축을 위해서는 원본 PDF 문서를 검색 및 LLM 컨텍스트 구성에 적합한 형태로 가공하는 것이 매우 중요합니다.

**1. 문서 전처리 (Preprocessing)**

청킹에 앞서 다음과 같은 전처리 단계를 수행합니다.

* **텍스트 추출 (Text Extraction):**
  + **품질 확인:** PDF에서 텍스트를 정확하게 추출하는 것이 가장 중요합니다. 이미지로 된 텍스트가 있다면 OCR(광학 문자 인식) 도구를 사용하여 텍스트로 변환해야 합니다.
  + **구조 유지:** 텍스트 추출 시 원본 문서의 구조(단락, 제목, 목록, 표 등)를 최대한 유지해야 합니다. 이는 이후 의미론적 청킹에 도움이 됩니다. Python의 PyMuPDF (fitz)나 pdfminer.six 같은 라이브러리를 활용할 수 있습니다.
  + **불필요한 요소 제거:** 머리글(header), 바닥글(footer), 페이지 번호 등 검색이나 내용 이해에 불필요한 반복적 요소는 제거합니다.
* **텍스트 정제 (Text Cleaning):**
  + **줄 바꿈 문자 정규화:** 여러 줄 바꿈 문자를 단일화하거나, 문장 중간의 불필요한 줄 바꿈을 제거합니다.
  + **특수 문자 처리:** 깨진 문자나 인코딩 오류를 수정합니다.
  + **공백 정규화:** 여러 개의 공백을 하나로 줄입니다.

**2. 청킹 알고리즘 구성 (Chunking Algorithm)**

의미론적 경계를 유지하면서 적절한 크기로 문서를 분할하는 것이 핵심입니다. NOSS 문서의 구조적 특성을 고려한 청킹 전략은 다음과 같습니다.

* **최상위 계층: 역량 단위 (Competency Unit - CU) 식별**
  + 각 CU는 독립적인 정보 단위를 형성하므로, CU별로 정보를 그룹화하는 것이 좋습니다.
  + PDF 내에서 "COMPETENCY UNIT TITLE", "CU CODE" 등의 패턴을 사용하여 각 CU의 시작과 끝을 식별합니다. (예: "15.1. Instructional Delivery Implementation" )
  + CU의 기본 정보(CU 코드, 제목, 설명, 학습 성과 등)는 그 자체로 중요한 청크가 될 수 있습니다.
* **중간 계층: 작업 활동 (Work Activity - WA) 식별**
  + 각 CU 내에는 여러 WA가 존재합니다. WA는 CU의 하위 정보 단위입니다.
  + "WORK ACTIVITIES"와 같은 제목이나 번호 매기기(예: "1. Analyse training requirements.")를 기준으로 각 WA 섹션을 구분합니다.
  + WA의 제목과 개요 설명(있다면)을 하나의 청크로 만들 수 있습니다.
* **하위 계층: WA 내 세부 항목 (Related Knowledge, Related Skills, Performance Criteria 등) 청킹**
  + **의미 기반 청킹 (Semantic Chunking):**
    - **단락(Paragraph) 기준:** 가장 일반적인 방법입니다. 각 단락을 하나의 청크로 만듭니다. NOSS PDF에서 "Related Knowledge", "Related Skills", "Performance Criteria", "Assessment Criteria" 등의 하위 섹션들은 주로 설명적인 단락으로 구성되어 있습니다.
    - **목록(List) 항목 기준:** 불릿 포인트나 번호 매기기 목록은 각 항목을 개별 청크로 만들거나, 관련성이 높은 항목들을 그룹화하여 하나의 청크로 만들 수 있습니다. 예를 들어, "Performance Criteria" 밑의 각 기준 항목은 개별적인 의미를 가집니다.
    - **고정 크기 청킹 (Fixed-size Chunking) - 보조적 사용:** 의미론적 경계가 모호하거나 매우 긴 단락의 경우, 토큰 수(예: 200-500 토큰)를 기준으로 분할할 수 있습니다. 하지만 이 경우 문맥이 잘릴 수 있으므로, \*\*중첩(Overlap)\*\*을 사용하여 청크 간의 연속성을 유지하는 것이 좋습니다. (예: 50 토큰 중첩)
  + **표(Table) 데이터 청킹:**
    - NOSS 문서 내 표(예: "Tools, Equipment and Materials (TEM)" , "Competency Weightage" )는 특별한 처리가 필요합니다.
    - **행(Row) 기준 청킹:** 각 행을 하나의 청크로 만들고, 열 제목을 메타데이터로 포함하거나 텍스트 내용에 자연스럽게 기술합니다. (예: "항목: Whiteboard, 비율: 1:25, CU: C01, C02, ...")
    - **전체 표 요약 청킹:** 표 전체를 설명하는 요약 텍스트를 생성하여 청크로 만들 수도 있습니다.
    - 마크다운(Markdown) 변환: 표를 마크다운 형식으로 변환하여 LLM이 더 잘 이해하도록 할 수 있습니다.
* **메타데이터 부여 (Metadata Assignment):**
  + 각 청크 생성 시, 해당 청크의 출처와 문맥을 명확히 하는 메타데이터를 풍부하게 부여합니다. (이전 답변의 "벡터 스토어 스키마" 참고)
    - noss\_code, cu\_code, cu\_title, wa\_id (예: "C01-WA1"), wa\_title
    - section\_type (예: "RelatedKnowledge", "PerformanceCriteria", "TEM\_List")
    - 페이지 번호
    - 원본 문서명
  + **계층적 메타데이터:** 상위 CU 정보, WA 정보를 하위 청크에 상속하거나 연결하여 검색 시 관련 정보를 함께 가져올 수 있도록 합니다.

**3. 청킹 알고리즘 구성 방안 (예시적 접근)**

1. **문서 로드 및 기본 정제:** PDFMiner, PyMuPDF 등을 사용하여 텍스트와 기본 구조(글꼴 크기, 위치 등) 추출. 머리글/바닥글 제거.
2. **최상위 구조 분할 (CU 단위):**
   * "COMPETENCY UNIT TITLE" 또는 "CU CODE P853-..." 패턴을 찾아 CU 경계 식별.
   * CU별 텍스트 블록 분리.
3. **CU 내 WA 단위 분할:**
   * 각 CU 블록 내에서 "WORK ACTIVITIES"와 그 하위 번호(1., 2., a), b) 등)를 기준으로 WA 경계 식별.
   * WA별 텍스트 블록 분리.
4. **WA 내 세부 내용 청킹:**
   * WA 블록 내에서 "Related Knowledge", "Related Skills", "Performance Criteria" 등의 부제목을 기준으로 1차 분할.
   * 각 부제목 하위 내용을 단락, 목록 항목 단위로 분할. (Langchain의 RecursiveCharacterTextSplitter 등을 활용하되, NOSS 문서의 특성을 반영한 구분자 설정)
   * 표 데이터는 별도 로직으로 행 단위 또는 의미 단위로 분할/변환.
5. **청크 크기 및 중첩 조절:**
   * 생성된 청크의 토큰 수를 확인. 너무 길면 추가 분할 (의미 유지하며), 너무 짧으면 인접 청크와 병합 고려.
   * 필요시 청크 간 중첩 적용.
6. **메타데이터 생성 및 연결:** 각 청크에 해당하는 noss\_code, cu\_code, wa\_id, section\_type 등의 메타데이터를 추출하거나 규칙 기반으로 생성하여 부여.

**4. 추가 고려 사항**

* **반복적 개선:** 초기 청킹 전략을 적용 후, 실제 검색 결과와 LLM의 답변 품질을 보면서 청킹 단위, 크기, 메타데이터 부여 방식을 지속적으로 개선해야 합니다.
* **다양한 청킹 전략 혼합:** 문서의 부분별 특성에 따라 다른 청킹 전략을 혼합하여 사용할 수 있습니다. (예: 개요 부분은 큰 청크, 세부 지식 부분은 작은 청크)
* **시각 정보 활용:** PDF 내 다이어그램이나 이미지가 중요한 정보를 담고 있다면, 이를 텍스트로 설명하거나 이미지 캡셔닝 모델을 활용하여 텍스트화하는 방안도 고려할 수 있습니다 (이 경우 복잡도 증가).
* **Langchain 등 프레임워크 활용:** Langchain과 같은 LLM 애플리케이션 개발 프레임워크는 다양한 텍스트 분할기(Text Splitter)와 문서 로더(Document Loader)를 제공하므로, 이를 활용하여 효율적으로 청킹 파이프라인을 구축할 수 있습니다.

이러한 전처리 및 청킹 전략을 통해 NOSS PDF 문서의 정보를 RAG 시스템이 효과적으로 이해하고 활용할 수 있는 형태로 만들 수 있습니다. 📚➡️🧩

소스

그럼이러한 시스템 구축을 위한 기술아키텍처를 제안해 주세요

생각하는 과정 표시

네, 제안해주신 NOSS 문서 자동 생성 시스템 구축을 위한 기술 아키텍처를 제안해 드립니다. 이 아키텍처는 데이터 수집부터 최종 문서 생성까지의 과정을 지원하며, RAG (Retrieval Augmented Generation) 모델을 핵심으로 활용하는 방안을 포함합니다.

**시스템 기술 아키텍처 제안**

이 아키텍처는 여러 계층으로 구성되어 각 기능을 모듈화하고, 필요에 따라 기술 스택을 유연하게 선택할 수 있도록 합니다.

코드 스니펫

graph LR

subgraph Presentation Layer

UI[User Interface (Web/Desktop)]

end

subgraph Application & Orchestration Layer

API[API Gateway / Backend Server]

WF[Workflow Orchestrator]

end

subgraph RAG Core & AI Layer

QM[Query Manager / Prompt Engine]

LLM[Large Language Model (LLM)]

RT[Retrieval Engine]

end

subgraph Data & Storage Layer

VS[Vector Store (Embeddings & Metadata)]

SDS[Structured Data Store (Excel Parsed Data)]

FS[File Store (Original PDFs, Templates)]

EM[Embedding Model Service]

end

subgraph Data Ingestion & Preprocessing Layer

DP[Document Processor (Parser, Cleaner, Chunker)]

end

UI -- HTTP/WebSocket --> API

API -- Triggers --> WF

WF -- Manages --> QM

QM -- Sends Query + Context --> LLM

QM -- Requests Chunks --> RT

RT -- Searches --> VS

RT -- Queries --> SDS

LLM -- Generates Content --> QM

WF -- Handles Output --> OutputGen[Output Generator / Template Filler]

OutputGen -- Produces --> FinalDoc[Generated Document]

OutputGen -- Uses --> FS[Word Templates]

DP -- Ingests --> FS[Source PDFs, Excels]

DP -- Sends Chunks for Embedding --> EM

EM -- Returns Embeddings --> DP

DP -- Stores Chunks & Metadata --> VS

DP -- Stores Parsed Excel Data --> SDS

**1. 데이터 계층 (Data & Storage Layer)**

이 계층은 시스템 운영에 필요한 모든 데이터를 저장하고 관리합니다.

* **파일 저장소 (File Store / Document Store):**
  + **목적:** 원본 NOSS PDF 문서, 엑셀 계획 자료, 워드 문서 템플릿 등을 저장합니다.
  + **기술 예시:** AWS S3, Google Cloud Storage, Azure Blob Storage, 로컬 파일 시스템 (NAS 등)
* **벡터 저장소 (Vector Store):**
  + **목적:** 문서에서 청킹된 텍스트의 임베딩 벡터와 관련 메타데이터를 저장하여 빠르고 효율적인 유사도 검색을 지원합니다.
  + **기술 예시:**
    - **전문 벡터 데이터베이스:** Pinecone, Weaviate, Milvus, Qdrant, ChromaDB
    - **벡터 검색 지원 데이터베이스:** PostgreSQL (pgvector 확장), Elasticsearch/OpenSearch (k-NN 기능)
* **구조화된 데이터 저장소 (Structured Data Store):**
  + **목적:** 파싱된 엑셀 데이터(예: WIM 제목, 교육 시간, 특정 학습 목표 등)를 저장하여 정확한 값 조회 및 필터링에 사용합니다.
  + **기술 예시:** PostgreSQL, MySQL, SQLite (간단한 경우), NoSQL 데이터베이스 (MongoDB - JSON 형태 저장 시)
* **임베딩 모델 서비스 (Embedding Model Service):**
  + **목적:** 텍스트 청크를 벡터 임베딩으로 변환합니다.
  + **기술 예시:**
    - **상용 모델 API:** OpenAI Embeddings (예: text-embedding-ada-002, text-embedding-3-small), Google Vertex AI Embeddings
    - **오픈소스 모델 (자체 호스팅 또는 Hugging Face 등 활용):** Sentence-BERT, BGE, multilingual-e5-large 등 한국어 지원 모델

**2. 데이터 수집 및 전처리 계층 (Data Ingestion & Preprocessing Layer)**

소스 문서를 시스템에서 활용 가능한 형태로 가공하는 단계입니다.

* **문서 처리기 (Document Processor):**
  + **목적:** 원본 문서(PDF, Excel)를 로드하여 텍스트/데이터 추출, 정제, 청킹 작업을 수행하고, 임베딩을 위해 임베딩 모델 서비스로 전달 후 결과를 벡터 스토어 및 구조화된 데이터 저장소에 저장합니다.
  + **기술 예시:**
    - **PDF 처리:** Python 라이브러리 (PyMuPDF/fitz - 텍스트 및 구조 추출, pdfminer.six), OCR (Tesseract - 이미지 기반 PDF용)
    - **Excel/CSV 처리:** Python pandas 라이브러리
    - **청킹 로직:** Langchain (RecursiveCharacterTextSplitter, SemanticChunker), Spacy (문장 분리 등), 또는 NOSS 문서 구조에 특화된 커스텀 Python 스크립트
    - **파이프라인 관리:** Apache Airflow, Prefect (주기적/자동화된 문서 처리 시)

**3. RAG 핵심 및 AI 계층 (RAG Core & AI Layer)**

사용자의 요청을 받아 정보를 검색하고 내용을 생성하는 핵심 두뇌 역할을 합니다.

* **검색 엔진 (Retrieval Engine):**
  + **목적:** 사용자 질의(또는 시스템 내부 생성 질의)를 바탕으로 벡터 저장소에서 관련성이 높은 텍스트 청크를 검색하고, 구조화된 데이터 저장소에서 필요한 정보를 조회합니다. 메타데이터 필터링 기능을 포함합니다.
  + **기술 예시:** 선택한 벡터 저장소의 SDK/API, Langchain/LlamaIndex의 Retriever 모듈
* **질의 관리자 / 프롬프트 엔진 (Query Manager / Prompt Engine):**
  + **목적:** 사용자의 요청을 LLM이 이해할 수 있는 프롬프트로 변환하고, 검색 엔진을 통해 얻은 컨텍스트(검색된 청크, 조회된 데이터)와 함께 LLM에 전달합니다.
  + **기술 예시:** Langchain (Prompt Templates, LCEL), LlamaIndex (Query Engines), 커스텀 Python 로직
* **대규모 언어 모델 (Large Language Model - LLM):**
  + **목적:** 제공된 프롬프트와 컨텍스트를 기반으로 최종 문서에 들어갈 텍스트 내용을 생성합니다.
  + **기술 예시:**
    - **상용 모델 API:** OpenAI (GPT-4, GPT-3.5-turbo), Anthropic (Claude 시리즈), Google (Gemini)
    - **오픈소스 모델:** Llama, Mistral 등 (자체 호스팅 또는 API 제공 플랫폼 활용)

**4. 애플리케이션 및 오케스트레이션 계층 (Application & Orchestration Layer)**

사용자 요청 처리, 비즈니스 로직 수행, 각 계층 간의 작업 흐름을 조정합니다.

* **API 게이트웨이 / 백엔드 서버 (API Gateway / Backend Server):**
  + **목적:** 사용자 인터페이스로부터 요청을 받아 처리하고, 핵심 로직을 실행하며, 응답을 반환합니다.
  + **기술 예시:** Python 기반 프레임워크 (FastAPI, Flask, Django), Node.js (Express)
* **워크플로우 오케스트레이터 (Workflow Orchestrator):**
  + **목적:** 문서 생성 요청에 따른 여러 단계(정보 검색, LLM 호출, 결과 조합 등)의 복잡한 작업 흐름을 관리하고 조율합니다.
  + **기술 예시:** Langchain Expression Language (LCEL), LlamaIndex Pipelines, Celery (비동기 장기 작업 처리 시)
* **출력 생성기 / 템플릿 필러 (Output Generator / Template Filler):**
  + **목적:** LLM이 생성한 텍스트를 최종 출력 형태(구조화된 텍스트, 워드 파일 등)로 가공합니다.
  + **기술 예시:** Python python-docx (워드 파일 생성/수정 시), Jinja2 (텍스트 기반 템플릿 사용 시), 또는 단순 JSON/Markdown 포맷터

**5. 프레젠테이션 계층 (Presentation Layer)**

사용자가 시스템과 상호작용하는 인터페이스입니다.

* **사용자 인터페이스 (User Interface):**
  + **목적:** 사용자가 문서를 요청하고, 파라미터를 입력하며, 생성된 결과를 확인할 수 있는 화면을 제공합니다. 관리자는 문서 및 시스템 설정을 관리합니다.
  + **기술 예시:**
    - **웹 프론트엔드:** React, Vue.js, Angular
    - **빠른 프로토타이핑:** Streamlit, Gradio (Python 기반)
    - **데스크톱 애플리케이션:** (필요시) Qt (PyQt/PySide), Electron

**기술 선택 시 고려 사항:**

* **데이터의 특성 및 양:** 관리할 문서의 양, 구조의 복잡성
* **성능 요구사항:** 검색 속도, 문서 생성 시간
* **개발팀의 기술 숙련도:** 익숙한 프로그래밍 언어 및 프레임워크
* **비용:** 클라우드 서비스 사용료, 상용 LLM API 비용, 오픈소스 활용 시 인프라 비용
* **확장성 및 유지보수:** 향후 기능 추가 및 시스템 확장을 고려한 설계
* **보안:** 문서 접근 권한 및 데이터 보안

이 아키텍처는 일반적인 RAG 시스템 구성을 기반으로 제안된 것이며, 실제 구축 시에는 구체적인 요구사항과 제약 조건에 따라 각 컴포넌트의 기술 스택을 최적화해야 합니다.

네, 업로드해주신 엑셀 파일 (여러 개의 CSV 파일로 변환됨)을 분석했습니다. 이 파일들은 앞서 첨부해주신 PDF 문서 (P853-002-4:2021 TVET Instruction Level 4)에서 언급된 특정 항목들을 추출하거나 계획하는 데 사용되는 기준 정보 및 세부 계획 양식들로 보입니다.

각각의 CSV 파일 (엑셀 시트)은 다음과 같은 내용을 담고 있는 것으로 파악됩니다:

1. **PENJAJARAN CU (역량 단위 정렬)**: 국가직업능력표준(NOSS) 역량 단위(CU)에 대한 메타데이터(섹션, 그룹, 영역, NOSS 명칭, CU 명칭) 및 "COPTPA"(PDF에서도 언급된 TVET 프로그램 인증 실행 규약)에 따른 교육 기간 정렬 관련 정보를 포함하는 것으로 보입니다.
2. **PENJAJARAN WA (작업 활동 정렬)**: 특정 역량 단위(예: C01 - 교육 전달 실행) 하위의 작업 활동(WA) 목록, 작업 활동 가중치/시간, COPTPA에 따른 교육 시간 정렬 정보를 포함합니다.
3. **JPPMAPP (학습 활동 및 평가 시간 비율 결정 일정)**: 역량 단위에 대한 교육 시간을 대면 학습(Bersemuka), 자기 주도 학습(Kendiri), 평가(Penilaian) 등 다양한 학습 모드별 비율과 시간으로 세분화하는 일정 또는 템플릿으로 보입니다.
4. **PENGIRAAN RK&RS (COPTPA 정렬 시간에 따른 대면 학습 시간(RK & RS) 계산)**: 작업 활동(WA) 및 COPTPA 정렬 시간을 기준으로 관련 지식(RK) 및 관련 기술(RS)에 대한 대면 교육 시간을 계산하는 것으로 보입니다 (예: RK 40%, RS 60%로 분류).
5. **JPMPBT (교육 기간 표 - 이론)**: 이론 구성 요소에 대한 작업 활동, 수업 계획 이론(LPT 명칭), 이론 기간 및 학습 목표를 나열합니다.
6. **JPMPBP (교육 기간 표 - 실습)**: 실습 구성 요소에 대한 작업 활동, 수업 계획 실습(LPP 명칭), 실습 기간 및 학습 목표를 나열합니다.
7. **JPW (TEORI) (필기 교육 자료 배포 일정 - 이론)**: 작업 활동을 이론 기반 필기 교육 자료(예: 교수 계획, 정보 시트, 과제 시트)에 매핑하는 일정으로 보입니다.
8. **JPW (AMALI) (필기 교육 자료 배포 일정 - 실습)**: 작업 활동을 실습 구성 요소(예: 수업 계획, 작업 시트)에 매핑하는 일정입니다.

이러한 엑셀 파일의 내용들은 PDF 문서에서 설명하는 NOSS 표준의 구현 세부 사항을 계획하고 문서화하는 데 사용되는 구체적인 양식 및 계산 방법들로 보입니다. 특히 PDF 문서의 역량 단위 중 "P853-002-4:2021-C02 필기 교육 자료(WIM) 개발" 부분에서 이러한 문서들(JPW, JPPMP, JPMPB 등)의 개발 필요성을 언급하고 있습니다. 엑셀 시트들은 바로 이 문서들의 템플릿 또는 예시로 활용될 수 있습니다.

네, 워드 문서 양식 "C01-KK(1-6)-ENG FORMAT.docx"에 대해 분석해 드리겠습니다.

제가 직접 .docx 파일의 내부 구조(제목 스타일, 표, 특정 필드 등)를 열어보거나 파싱할 수는 없습니다. 하지만 파일명과 이전에 논의된 문서들(NOSS PDF, 엑셀 계획 자료)의 맥락을 통해 이 문서의 목적과 예상 내용을 추론해 볼 수 있습니다.

**파일명 "C01-KK(1-6)-ENG FORMAT.docx" 분석:**

* **C01:** NOSS 표준 문서 P853-002-4:2021의 역량 단위(Competency Unit) \*\*C01 "Instructional Delivery Implementation"\*\*과 관련된 문서일 가능성이 매우 높습니다.
* **KK:** 말레이시아 NOSS/TVET 맥락에서 "KK"는 주로 \*\*"Kertas Kerja" (Worksheet 또는 Work Paper)\*\*를 의미합니다. 이는 학습자가 특정 작업 활동과 관련된 실제적인 단계나 과제를 수행하도록 안내하는 문서입니다.
* **(1-6):** 이는 C01 역량 단위 내의 작업 활동(Work Activity) 1부터 6까지를 의미하거나, 해당 "Kertas Kerja" 문서 내의 1번부터 6번까지의 섹션 또는 과제를 나타낼 수 있습니다. NOSS PDF에 따르면 C01 "Instructional Delivery Implementation"에는 5개의 작업 활동이 정의되어 있습니다:
  1. Analyse training requirements.
  2. Prepare pre training requirement.
  3. Conduct training preparation.
  4. Deliver training presentation.
  5. Conduct training confirmation. 따라서 "(1-6)"은 이 5가지 작업 활동을 더 세분화한 6개의 주요 섹션이나 과제를 포함하는 양식일 수 있습니다.
* **ENG FORMAT:** 해당 문서 양식이 영어로 작성되었음을 나타냅니다.

**예상되는 문서의 목적 및 내용:**

이 문서는 C01 "Instructional Delivery Implementation" 역량 단위에 대한 **실습 워크시트(Kertas Kerja) 템플릿**일 것으로 보입니다. 주요 목적은 학습자(또는 사용자)가 C01의 각 작업 활동(Work Activities)과 관련된 실질적인 작업이나 과제를 수행하고 기록할 수 있도록 안내하는 것입니다.

포함될 수 있는 예상 내용은 다음과 같습니다:

* **작업 활동 1 (Analyse training requirements):** 교육 요구사항 분석을 위한 질문, 템플릿, 또는 단계별 가이드
* **작업 활동 2 (Prepare pre training requirement):** 사전 교육 요구사항 준비를 위한 체크리스트 또는 절차 안내
* **작업 활동 3 (Conduct training preparation):** 교육 준비 수행을 위한 가이드라인 또는 고려 사항
* **작업 활동 4 (Deliver training presentation):** 교육 발표 계획을 위한 구조 또는 주요 항목 제시
* **작업 활동 5 (Conduct training confirmation):** 교육 확인 및 세션 평가 방법을 위한 지침 또는 양식

이러한 내용은 NOSS PDF 문서의 C01 "Curriculum of Competency Unit" 섹션에 명시된 **"RELATED SKILLS"** 및 \*\*"ASSESSMENT CRITERIA"\*\*와 밀접하게 연관될 것입니다. 또한, 엑셀 파일 중 "JPW (AMALI)" (실습용 WIM 배포 일정) 시트는 각 작업 활동에 해당하는 "WORK SHEET" (Kertas Kerja와 동일)의 구체적인 제목을 제공할 수 있습니다.

**시스템 구축과의 연관성:**

이 워드 문서는 사용자께서 구축하고자 하는 RAG 기반 문서 자동 생성 시스템에서 **출력 템플릿** 중 하나로 사용될 것입니다. 시스템은 NOSS PDF와 엑셀 계획 자료에서 관련 정보를 추출하고, LLM을 통해 내용을 생성한 후, 이 "C01-KK(1-6)-ENG FORMAT.docx" 템플릿의 해당 섹션에 맞춰 채워 넣는 방식으로 문서를 자동 생성하게 됩니다.

**다음 단계:**

이 워드 템플릿을 시스템에서 활용하거나 제가 내용을 채우는 데 도움을 드리려면, \*\*문서의 구체적인 내부 구조 (예: 주요 섹션 제목, 표의 열 제목, 채워져야 할 항목 목록 등)\*\*를 알려주시거나 문서의 텍스트 내용을 제공해주시면 더 정확한 지원이 가능합니다.

네, 다시 한번 "01. C01-KP(1-9)PM-ENG FORMAT(1).docx" 문서를 분석 요청하셨네요.

이전에 이 파일명과 제공해주신 다른 문서들(NOSS PDF, 엑셀 자료)의 맥락을 바탕으로 분석해 드린 바 있습니다. 제가 .docx 파일의 내부 콘텐츠나 구체적인 구조를 직접 열어볼 수는 없다는 점을 다시 한번 말씀드립니다.

하지만 파일명을 통해 다음과 같은 내용을 다시 한번 유추해 볼 수 있습니다:

* **C01:** NOSS 표준 P853-002-4:2021의 역량 단위(Competency Unit) \*\*C01 "Instructional Delivery Implementation"\*\*과 관련된 문서로 보입니다.
* **KP:** "Kertas Penerangan" (Information Sheet - 정보 시트)의 약자일 가능성이 높습니다. 이는 NOSS PDF에서 언급된 필기 교육 자료(WIM) 중 하나입니다.
* **PM:** "Pelan Mengajar" (Teaching Plan/Lesson Plan - 교수 계획)의 약자일 수 있습니다. 이 또한 NOSS PDF에서 언급된 WIM 중 하나입니다.
  + 파일명에 "KP"와 "PM"이 함께 있는 것으로 보아, 이 템플릿은 정보 시트(KP)와 교수 계획(PM)을 함께 작성하거나, 한 문서가 다른 문서의 내용을 참조하여 작성될 때 사용되는 통합적인 양식일 수 있습니다. 또는 "KP"가 주된 문서 유형이고 "PM"은 관련 교수 계획을 참조한다는 의미일 수도 있습니다.
* **(1-9):** 이 문서 템플릿 내에 1번부터 9번까지의 주요 섹션이나 항목이 있음을 나타낼 수 있습니다. 정보 시트나 교수 계획에 일반적으로 포함되는 항목들의 개수와 관련될 수 있습니다.
* **ENG FORMAT:** 영어로 된 양식임을 의미합니다.

**예상되는 문서의 목적:**

이 문서는 C01 역량 단위에 대한 **정보 시트(Kertas Penerangan)** 또는 **교수 계획(Pelan Mengajar)**, 혹은 이 두 가지를 포괄하는 교육 자료를 작성하기 위한 템플릿일 가능성이 큽니다.

* **정보 시트 (Kertas Penerangan)의 경우:** C01의 각 작업 활동(Work Activity)에 대한 "Related Knowledge" (관련 지식)를 중심으로 이론적인 내용을 상세히 설명하는 문서가 될 것입니다.
* **교수 계획 (Pelan Mengajar)의 경우:** C01의 교육 목표, 내용, 방법, 평가 등을 체계적으로 계획한 문서가 될 것입니다.

**이전 분석과 비교하여 어떤 점에 대해 더 자세한 분석을 원하시는지, 또는 이 문서를 가지고 어떤 작업을 하고 싶으신지 알려주시면 더 구체적인 도움을 드릴 수 있습니다.**

예를 들어, 이 템플릿의 특정 섹션(예: "Learning Objectives" 섹션)에 어떤 정보를 채워야 하는지, 또는 이 템플릿이 PDF나 엑셀의 어떤 부분과 연관되는지에 대해 궁금하신가요?