국민대학교
소프트웨어융합대학
소프트웨어학부

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	*LLMong*
팀 명	*Durmon:t*
문서 제목	결과보고서

Version	1.0
Date	2026-MAY-20

팀원	김 동연 (조장)
	강 아영
	김 동진
	박 가현
	배 경준
	하 승준

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 다학제간캡스톤디자인I 수강 학생 중 프로젝트 "LLMong"를 수행하는 팀 "Durmon:t"의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 "Durmon:t"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	결과보고서-LLMong.docx
원안작성자	김동연, 강아영, 김동진, 박가현, 배승준, 하승준
수정작업자	김동연, 강아영, 김동진, 박가현, 배승준, 하승준

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2026-05-15	김동연	1.0	최초 작성	
2026-05-20	박가현	2.0	전체 항목	전체 내용 수정, 흐름 정리 및 최종 확인

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

목 차

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

1 개요

1.1 프로젝트 개요

LLMong은 국내 업무 환경에서 널리 사용되는 문서를 AI가 활용 가능한 데이터로 변환하는 문서 파싱 서비스이다. 사용자는 웹에서 문서를 업로드하고 변환을 진행할 수 있으며, 시스템은 문서에서 텍스트, 표, 이미지, 메타데이터 등을 추출하여 검색과 질의응답에 활용할 수 있는 구조화 결과를 제공한다.

LLMong의 주요 서비스는 다음과 같다.

첫째, 문서 파싱 및 결과 검수 기능이다. 사용자는 HWP/HWPX, PDF, 이미지, Excel 등 다양한 형식의 문서를 업로드하고, 변환된 결과를 웹 화면에서 확인할 수 있다. 시스템은 문서 내 텍스트, 표, 이미지, 메타데이터를 추출하고 이를 Markdown, HTML table, JSON 등의 형태로 정리한다. 변환 완료 후에는 원본 문서와 추출 결과를 나란히 비교할 수 있어, 표 구조가 제대로 유지되었는지, 텍스트가 누락되지 않았는지, 이미지나 체크박스와 같은 문서 요소가 올바르게 반영되었는지 검수할 수 있다.

둘째, RAG 기반 문서 질의응답 기능이다. 변환된 문서는 검색 가능한 구조화 데이터로 저장되며, 사용자는 해당 문서를 기반으로 자연어 질의를 수행할 수 있다. 문서의 분량이 많거나 사용자가 필요한 정보를 직접 찾기 어려운 경우, RAG 챗봇을 통해 문서 내용에 대한 질문을 입력하고 관련 답변을 확인할 수 있다. 이를 통해 긴 보고서, 행정문서, 계약서, 연구자료 등에서 필요한 정보를 빠르게 탐색할 수 있으며, 문서 기반 업무 검토 시간을 줄일 수 있다.

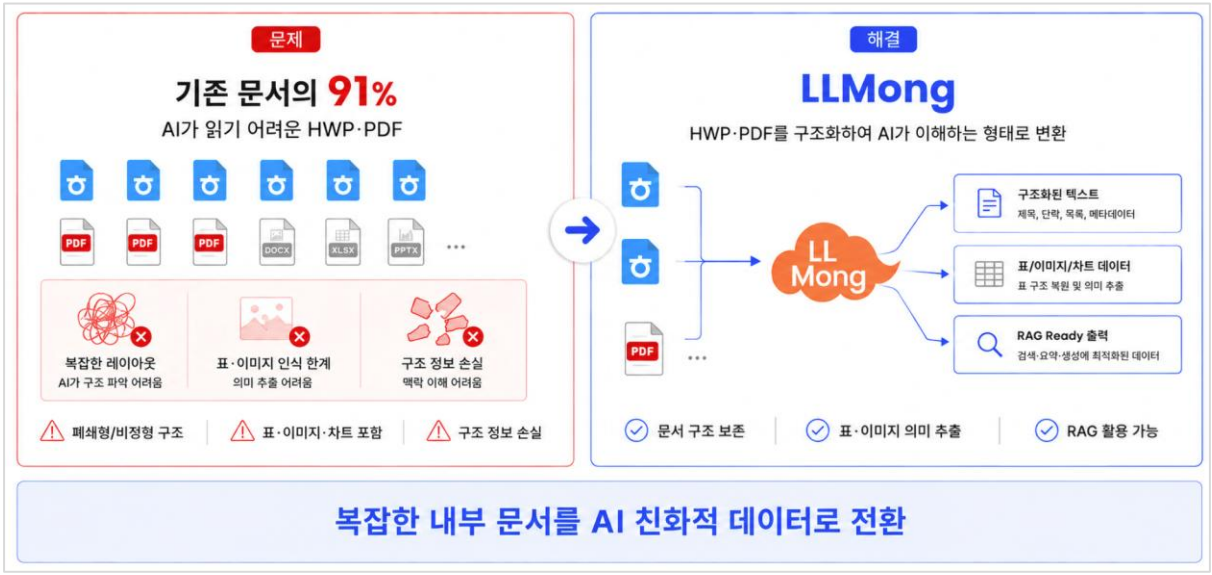
셋째, 온프레미스 LLM 처리 환경 지원이다. 공공기관, 금융기관, 연구기관과 같이 외부 클라우드 API 사용이 제한되는 환경을 고려하여, 로컬 GPU 서버 기반의 문서 분석 구조를 지원한다. 외부 API를 사용할 수 있는 환경에서는 OpenAI 또는 OpenRouter 기반 모델을 활용하고, 보안상 문서를 외부로 전송하기 어려운 환경에서는 Qwen 계열 로컬 모델을 활용할 수 있다. 이를 통해 민감한 문서를 내부망에서 처리할 수 있으며, 폐쇄망 환경에서도 AI 기반 문서 파싱 기능을 사용할 수 있다.

넷째, 서비스 대시보드 및 운영 모니터링 기능이다. 대시보드는 전체 문서 업로드 현황, 변환 작업 상태, 성공률, 실패 내역, 최근 작업 정보를 시각적으로 제공한다. 사용자는 현재 어떤 문서가 처리 중인지, 어떤 작업이 완료되었는지, 어떤 작업에서 오류가 발생했는지 한눈에 확인할 수 있다. 또한 에러 로그와 모니터링 정보를 통해 실패 원인을 파악하고, 문서 변환 파이프

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

라인의 안정성을 점검할 수 있다.

결과적으로 LLMong은 사람이 보관하던 문서를 AI가 검색하고 활용할 수 있는 지식 데이터로 전환한다. 단순 OCR이나 파일 변환 도구가 아니라, 문서 처리와 AI 활용을 연결하는 실사용형 문서 파싱 플랫폼을 목표로 개발되었다.



1.2 추진 배경 및 필요성

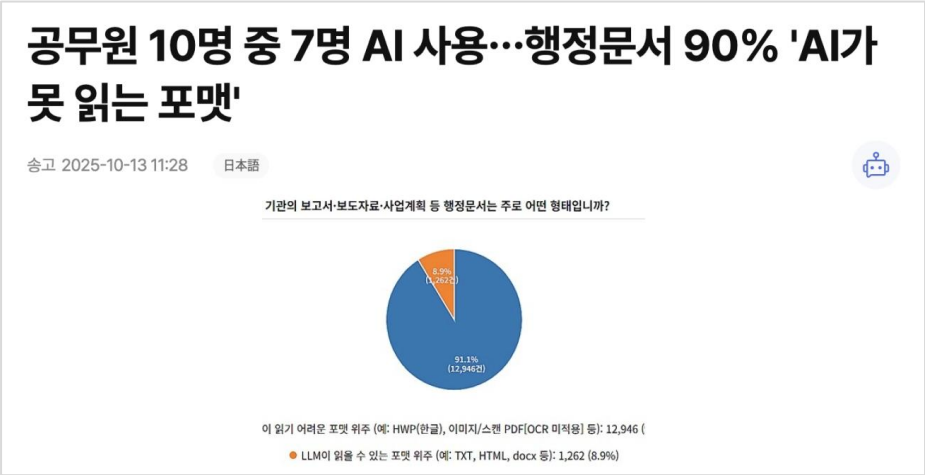
1.2.1 시장 현황

최근 GPT, Claude 등 대형 언어 모델(LLM)과 Vision-Language Model(VLM)의 발전으로 문서와 이미지에 포함된 정보를 AI가 직접 이해하고 분석하는 기술이 빠르게 고도화되고 있다. 기존 OCR 기술이 문자의 위치를 인식하고 텍스트를 추출하는 데 초점을 맞추었다면, 최신 VLM은 문서 이미지 안에 포함된 표 구조, 차트, 수식, 도형, 이미지 설명 등 복합적인 정보를 함께 해석할 수 있다는 점에서 차별성을 가진다. 특히 GPT-5.2, Qwen2.5-VL과 같은 모델은 단순 텍스트 추출을 넘어 문서의 시각적 구조와 문맥을 함께 분석할 수 있어, 문서 자동화 및 지식 관리 분야에서 활용 가능성이 커지고 있다.

그러나 실제 업무 환경에서 사용되는 문서는 AI가 바로 처리하기 쉬운 형태로 존재하지 않는 경우가 많다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

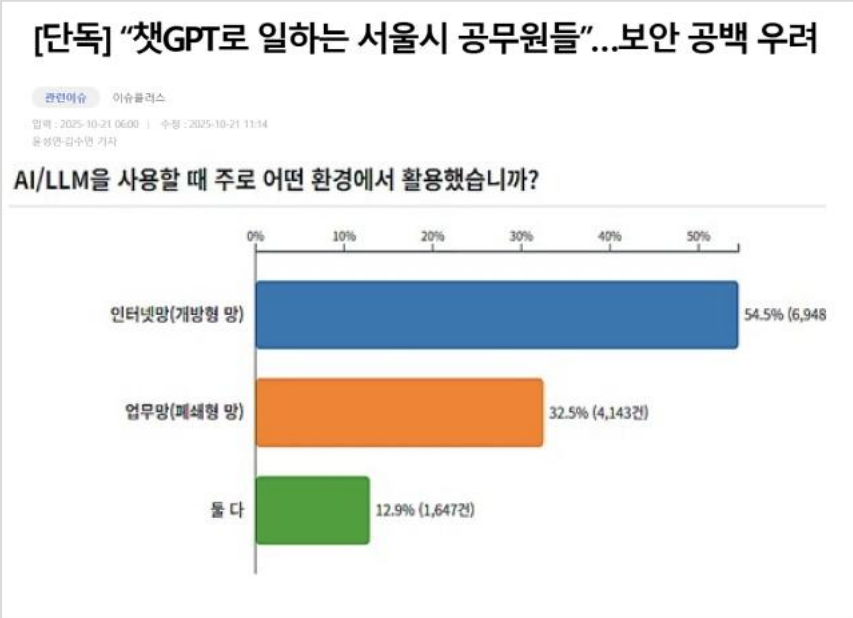
특히 국내 공공기관과 행정 조직에서는 HWP/HWPX 형식의 문서가 여전히 널리 사용되고 있으며, 기업 환경에서도 PDF, 스캔 이미지, Excel 기반 문서가 대량으로 생산되고 있다. 이러한 문서들은 사람이 보기에는 익숙하지만, AI 모델이나 RAG 시스템이 바로 활용하기에는 구조화 가 부족하다. 표, 차트, 이미지, 각주, 병합 셀, 다단 구성 등 복잡한 문서 요소가 포함된 경우 단순 OCR만으로는 정확한 의미와 구조를 추출하기 어렵다.



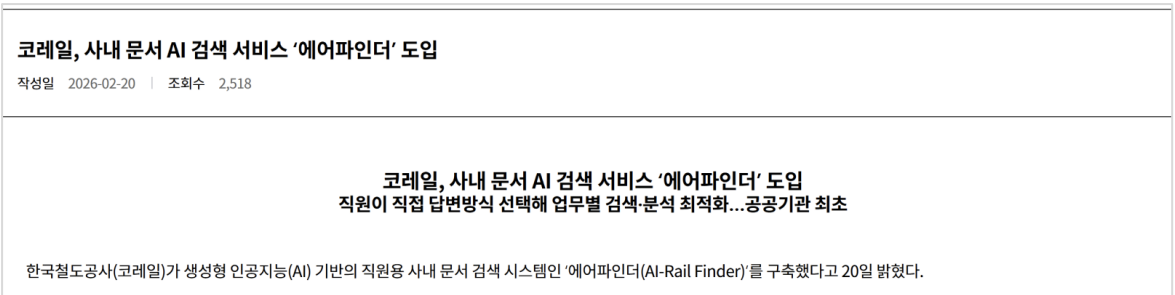
또한 공공·행정 문서의 상당수는 AI가 직접 읽기 어려운 포맷으로 관리되고 있다. 기사에서 언급된 바와 같이 행정문서의 상당 부분이 HWP, 이미지, 스캔 PDF 등 AI가 바로 해석하기 어려운 형태로 존재하며, 이는 문서 기반 AI 활용을 제한하는 주요 요인으로 작용한다. 즉, AI 기술 자체는 빠르게 발전하고 있으나, 실제 조직 내부의 문서 자산을 AI가 활용 가능한 데이터로 변환하는 전처리·파싱 기술은 여전히 중요한 과제로 남아 있다.

이와 함께 문서 기반 AI 활용에서는 보안 문제도 중요한 제한 요소로 작용한다. ChatGPT와 같은 외부 생성형 AI 서비스를 사용할 경우 문서 내용이 외부 클라우드 API로 전송될 수 있으며, 공공기관·금융기관·기업 내부 문서처럼 민감한 정보가 포함된 자료는 보안 정책상 외부 전송이 어렵다. 특히 행정문서, 계약서, 연구자료, 개인정보가 포함된 신청서 등은 외부 API 기반 처리만으로는 실제 도입에 한계가 있다. 따라서 문서 파싱 서비스는 클라우드 API 활용뿐만 아니라, 기관 내부 서버에서 모델을 직접 실행할 수 있는 온프레미스 처리 전략을 함께 고려해야 한다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

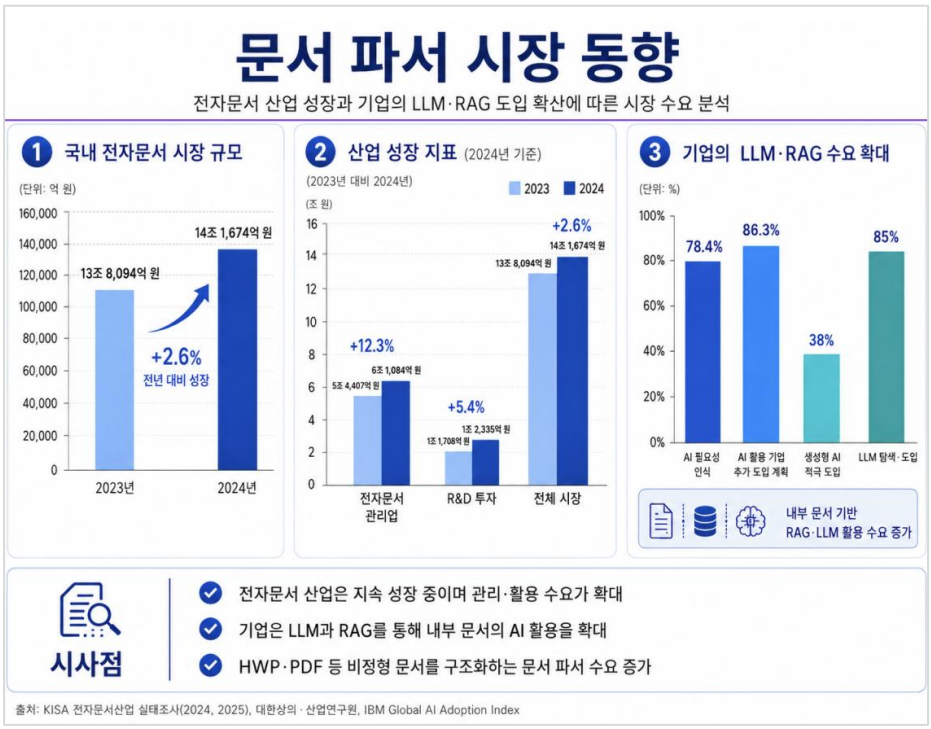


전자문서 시장 역시 지속적으로 성장하고 있다. 전자문서 산업 규모는 매년 증가하고 있으며, 기업들은 LLM과 RAG를 활용해 내부 문서를 검색하고 질의응답에 활용하려는 수요를 확대하고 있다. 특히 내부 규정, 보고서, 계약서, 사업계획서, 연구자료와 같은 비정형 문서를 AI가 검색 가능한 형태로 변환하려는 요구가 증가하고 있다. 이에 따라 단순 파일 저장이나 OCR을 넘어, 문서의 구조와 의미를 보존한 상태로 Markdown, HTML table, 메타데이터, 검색용 텍스트로 변환하는 문서 파서의 필요성이 커지고 있다.



이러한 시장 흐름 속에서 LLMong은 HWP/HWPX, PDF, 이미지, Excel 등 다양한 문서 형식을 AI 기반으로 분석하고, RAG 질의응답에 활용 가능한 구조화 데이터로 변환하는 것을 목표로 한다. 기존 OCR 중심의 문서 처리 방식이 텍스트 추출에 머물렀다면, LLMong은 표 구조 보존, 이미지 설명 생성, 메타데이터 관리, 문서 검색 및 질의응답 연계를 함께 제공함으로써 기업과 기관이 보유한 비정형 문서를 실제 AI 활용 자산으로 전환할 수 있도록 지원한다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.



따라서 본 프로젝트는 전자문서 산업의 성장, 기업의 LLM·RAG 도입 확대, 그리고 국내 HWP/PDF 중심 문서 환경이라는 시장적 요구를 반영한 시스템이다. 문서 파싱 기술은 향후 내부 지식 검색, 업무 자동화, 행정문서 분석, 보고서 자동 처리, 기업 문서 관리 시스템 등 다양한 분야로 확장될 수 있으며, LLMong은 이러한 수요에 대응하기 위한 AI 기반 문서 구조화 서비스로 개발되었다.

1.2.2 기존 솔루션의 한계

시장 한계	LLMONG의 해결
HWP/HWPX 지원 부족	HWP/HWPX 직접 파싱 및 PDF 변환 fallback 함께 지원
단순 텍스트 추출	텍스트, 표, 이미지, 차트 등 문서 요소를 구조화된 결과로 변환
표 구조 보존 어려움	HTML table, Markdown, JSON 메타데이터 형태로 표 구조 보존
대량 처리 비효율	Queue/Worker 기반 비동기 처리 구조로 대량 문서 처리 지원
공공기관 폐쇄망 수요	Qwen2.5-VL 7B 로컬 GPU 실행 및 Docker 기반 온프레미스 배포 지원

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

현재 시장에 존재하는 한국어 문서 처리 솔루션들은 다음과 같은 한계를 가지고 있다.

- HWP 지원 부족: 대부분의 문서 처리 솔루션은 PDF와 이미지 파일을 중심으로 지원하며, 국내 공공기관과 행정 문서에서 많이 사용되는 HWP/HWPX 파일의 구조적 변환 기능은 미흡하다. 특히 HWP 문서에 포함된 표, 문단, 이미지, 계층 구조를 원본에 가깝게 보존하는 데 한계가 있다.

툴	HWP 지원	지원 포맷
LlamaParse	X	PDF, DOCX, PPTX, 이미지
Docling (IBM)	X	PDF, DOCX, PPTX, XLSX, HTML, 이미지
AWS Textract	X	PDF, JPEG, PNG, TIFF
Google Document AI	X	PDF, 이미지

- 단순 텍스트 추출 중심: 기존 OCR 기반 솔루션은 문자를 인식하고 텍스트를 추출하는 데에는 강점이 있지만, 표 내부의 표, 병합 셀, 차트, 수식, 이미지 설명 등 복잡한 레이아웃 요소를 Markdown 또는 JSON과 같은 구조화된 형태로 변환하는 데에는 한계가 있다. 이로 인해 RAG나 LLM 기반 질의응답에서 문서의 의미 구조가 충분히 활용되지 못한다.

- 표 구조 인식 한계: 일부 솔루션은 일반적인 표 인식에서는 높은 정확도를 제공하지만, 중첩 표나 복잡한 행·열 구조를 가진 문서에서는 Markdown 변환 과정에서 표 구조가 붕괴될 수 있다. 특히 행 병합, 열 병합, 표 내부의 하위 표와 같은 요소는 단순 OCR 결과만으로 정확히 복원하기 어렵다.

- 대량 처리 비효율: 클라우드 API 기반 문서 처리 서비스는 페이지당 과금 구조를 가지는 경우가 많아, 수십만 장에서 수백만 장 규모의 문서를 처리할 때 비용 부담이 커진다. 또한 일부 온프레미스 솔루션은 도입과 구축에 긴 시간이 필요해 대량 문서를 빠르게 처리해야 하는 환경에는 적합하지 않다.

- 폐쇄망 환경 대응 한계: 공공기관, 금융기관, 연구기관처럼 보안이 중요한 환경에서는 문서를 외부 클라우드 API로 전송하기 어렵다. 그러나 기존 솔루션은 클라우드 API 의존도가 높거나 별도 엔터프라이즈 구축이 필요해, 폐쇄망 환경에서 즉시 활용하기 어렵다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

2 개발 내용 및 결과물

2.1 목표

본 프로젝트의 목표는 한국형 업무 문서를 생성형 AI와 RAG 시스템이 활용할 수 있는 구조화 데이터로 변환하는 AI 기반 Document Parser를 개발하는 것이다. 기존 문서 처리 방식이 단순히 문자 인식이나 파일 변환에 머물렀다면, 본 프로젝트는 문서의 내용뿐만 아니라 표, 이미지, 차트, 수식, 메타데이터와 같은 구조적 정보를 함께 추출하여 후속 AI 서비스에서 사용할 수 있는 형태로 제공하는 것을 목표로 한다.

국내 공공기관과 기업에서는 HWP/HWPX, PDF, 스캔 이미지, Excel 기반 문서가 여전히 대량으로 사용되고 있다. 이러한 문서는 사람이 읽기에는 익숙하지만, LLM이나 RAG 시스템이 바로 활용하기에는 구조화가 부족하다. 특히 표 내부의 표, 병합 셀, 이미지성 표, 차트, 수식, 문서 메타데이터와 같은 요소는 단순 OCR만으로 정확하게 처리하기 어렵다. 따라서 본 프로젝트는 다양한 문서 형식을 하나의 처리 흐름에서 분석하고, 검색과 질의응답에 적합한 데이터로 변환하는 것을 핵심 목표로 한다.

또한 문서 변환은 파일 크기와 페이지 수, AI 모델 호출 여부에 따라 처리 시간이 길어질 수 있으므로, 비동기 작업 큐와 Worker 구조를 적용하여 API 요청과 실제 변환 처리를 분리한다. 사용자는 작업이 진행되는 동안 WebSocket을 통해 상태를 확인할 수 있으며, 변환 완료 후 결과를 미리보기, 다운로드, RAG 질의응답에 활용할 수 있다.

보안이 중요한 환경도 고려하였다. 외부 AI API를 사용할 수 있는 환경에서는 OpenAI 또는 OpenRouter 기반 모델을 활용하고, 문서 외부 전송이 어려운 환경에서는 Qwen 계열 로컬 GPU 모델을 활용할 수 있는 실행 구조를 마련한다. 이를 통해 일반 클라우드 환경과 내부망 중심의 온프레미스 환경 모두에 대응 가능한 문서 파싱 시스템을 구축하는 것을 목표로 한다.

결과적으로 본 프로젝트는 국내 업무 문서를 AI 시대에 활용 가능한 지식 데이터로 전환하고, 문서 검색, 업무 자동화, 행정문서 분석, 기업 지식관리, RAG 기반 질의응답에 활용할 수 있는 기반 시스템을 개발하는 것을 목표로 한다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

2.1.1 세부 목표 및 결과물

1. 문서 처리 파이프라인 구축
- 다양한 입력 문서를 단일 파이프라인에서 처리할 수 있는 구조를 구현한다.
- HWP/HWPX 문서는 변환 및 직접 파싱 방식을 조합하여 텍스트, 표, 이미지 정보를 추출한다.
- PDF 문서는 페이지 단위로 텍스트 블록과 이미지 영역을 분석한다.
- 이미지성 표, 차트, 수식 등 일반 파서만으로 처리하기 어려운 요소는 VLM 기반 분석을 적용한다.

최종 산출물:
- 문서 처리 파이프라인 모듈
- HWP/HWPX/PDF 변환 및 추출 모듈
- VLM 기반 문서 구조 분석 모듈
- TXT, Markdown, HTML table, JSON 변환 결과

2. 프론트엔드 웹 애플리케이션 개발
- 사용자가 브라우저에서 문서를 업로드하고 변환 작업을 생성할 수 있는 웹 UI를 구현한다.
- 파일 선택, 폴더 선택, 드래그 앤 드롭, 멀티 파일 업로드 기능을 제공한다.
- 모델 선택, 병렬 처리 설정, 작업 진행 상태 확인, 결과 미리보기, 다운로드 기능을 제공한다.
- 대시보드, 파일 관리, RAG, 에러 로그 화면을 구현한다.

최종 산출물:
- React 기반 프론트엔드 소스 코드
- 문서 변환 UI
- 대시보드 UI
- 파일 관리 UI
- RAG 질의응답 UI
- 에러 로그 UI

3. 백엔드 API 및 실시간 상태 전달 기능 개발
- 문서 업로드, 변환 작업 생성, 작업 상태 조회, 결과 조회, 다운로드, 삭제 API를 구현한다.
- 모델 목록, 대시보드 통계, 모니터링, 에러 로그, RAG 질의응답 API를 제공한다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

- WebSocket을 활용하여 작업의 대기, 진행, 완료, 실패, 취소 상태를 프론트엔드에 전달한다.
- Swagger UI를 통해 API 명세를 확인할 수 있도록 한다.

최종 산출물:
- FastAPI 기반 백엔드 서버
- 기능별 REST API 라우터
- WebSocket 진행 이벤트 API
- Swagger API 문서

4. 비동기 작업 처리 및 저장 구조 구축
- RabbitMQ 기반 작업 큐를 통해 문서 변환 요청과 실제 처리 작업을 분리한다.
- Worker가 큐에서 작업을 가져와 문서 변환을 수행하도록 구현한다.
- Redis를 활용하여 작업 상태 캐시와 빠른 상태 조회를 지원한다.
- SQLite를 활용하여 사용자, 문서, 작업 상태, 변환 결과를 저장한다.
- 실패 작업 재시도와 작업 취소 기능을 제공한다.

최종 산출물:
- RabbitMQ 작업 큐 구조
- Worker 실행 모듈
- 작업 상태 관리 모듈
- Redis 상태 캐시 연동
- SQLite 기반 문서 및 작업 저장소

5. RAG 질의응답 기능 구현
- 변환된 문서를 검색 가능한 단위로 분할하고 embedding 기반 검색을 수행한다.
- 사용자의 질문과 관련된 문서 내용을 검색한 뒤 답변을 생성한다.
- 프론트엔드 RAG 화면과 백엔드 RAG API를 연동한다.
- 표와 문서 구조를 보존한 결과가 검색 품질 향상에 기여하도록 설계한다.

최종 산출물:
- RAG API
- 문서 검색 및 질의응답 모듈
- RAG 채팅 UI
- embedding 기반 검색 구조

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

6. 배포 및 운영 환경 구성

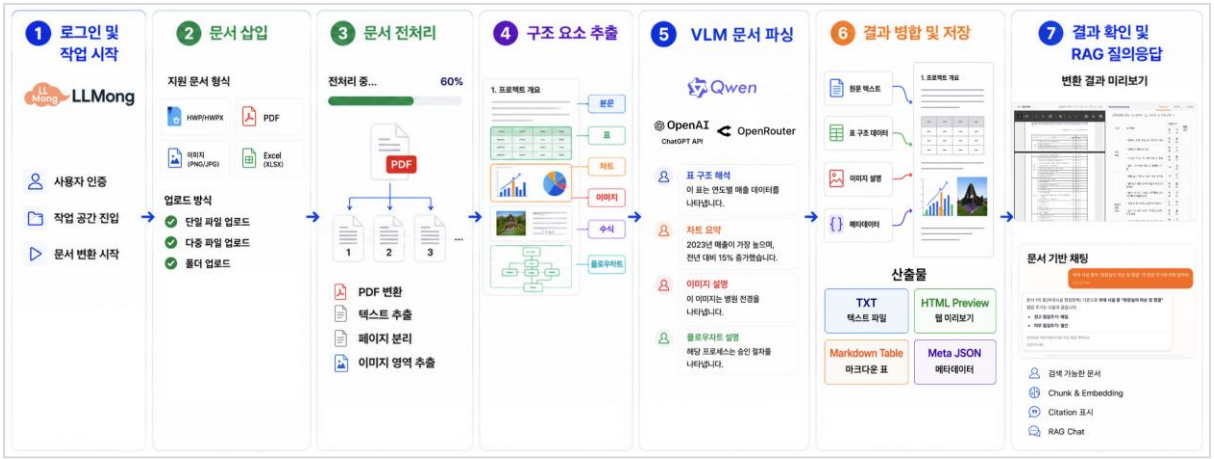
- Docker Compose 기반으로 Frontend, Backend, Worker, Redis, RabbitMQ를 실행할 수 있도록 구성한다.
- 일반 API 기반 실행 환경과 로컬 GPU 기반 실행 환경을 분리한다.
- Nginx를 통해 프론트엔드 정적 파일 제공과 API/WebSocket 프록시를 지원한다.
- GitHub Actions와 GHCR을 활용하여 빌드 및 배포 흐름을 구성한다.
- 운영자가 시스템 상태, 에러 로그, 작업 현황을 확인할 수 있도록 문서와 관리 기능을 제공한다.

최종 산출물:

- Dockerfile 및 Docker Compose 설정
- 일반 실행용 배포 환경
- 온프레미스 GPU 실행용 배포 환경
- 배포 가이드
- 운영자 매뉴얼

2.2 연구/개발 내용 및 결과물

2.2.1 연구/개발 내용



1. HWP/HWPX/PDF 문서 변환 파이프라인 구현

- HWP/HWPX, PDF, 이미지, Excel 등 다양한 문서 형식을 단일 문서 처리 파이프라인에서 처리하도록 구현하였다.
- HWP 문서는 pyhwp(hwp5html), LibreOffice, 텍스트 기반 추출 방식을 조합하여 변환 실패 가능성을 줄였다.
- HWPX 문서는 XML 구조를 분석하여 본문 텍스트, 표, 이미지 정보를 추출하도록 구현하였다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

- PDF 문서는 PyMuPDF를 활용하여 페이지 단위로 텍스트 블록, 이미지 영역, 표 구조를 추출하도록 구현하였다.
- 변환 과정에서 비정상적으로 생성되는 페이지나 OLE 바이너리 기반 쓰레기 페이지를 감지하여 결과 품질을 보완하였다.

2. VLM 기반 문서 구조 분석 기법 구현

- GPT 계열 모델, OpenRouter 기반 모델, Qwen 계열 모델을 활용할 수 있는 VLM 연동 구조를 구현하였다.
- 일반 파서만으로 분석하기 어려운 이미지성 표, 차트, 수식, 흐름도, 복합 레이아웃 영역을 VLM으로 분석하도록 구성하였다.
- 표, 이미지, 수식 등 문서 요소별 프롬프트를 설계하여 문서 구조를 보다 정확하게 추출할 수 있도록 하였다.
- 분석 결과는 일반 텍스트, Markdown, HTML table, JSON 메타데이터 형태로 제공되도록 구현하였다.
- HTML table과 Markdown table을 함께 제공하여 사람이 확인하기 쉬운 결과와 RAG 시스템이 활용하기 쉬운 결과를 모두 지원하였다.

3. React 기반 프론트엔드 웹 애플리케이션 구현

- React, TypeScript, Vite 기반의 SPA를 구현하였다.
- FSD(Feature-Sliced Design) 구조를 적용하여 app, pages, widgets, features, entities, shared 계층으로 프론트엔드 코드를 분리하였다.
- 문서 업로드, 파일/폴더 선택, 드래그 앤 드롭 업로드, 변환 설정, 작업 생성, 진행 상태 확인, 결과 미리보기, 다운로드 기능을 구현하였다.
- TanStack Router를 활용하여 화면 단위 라우팅을 구성하고, TanStack Query를 활용하여 API 요청, 캐싱, 자동 갱신을 처리하였다.
- Zustand를 활용하여 업로드 파일 목록, 변환 설정 등 클라이언트 상태를 관리하였다.
- Tailwind CSS, shadcn/ui, Radix UI 기반으로 대시보드, 문서 변환, 파일 관리, RAG, 에러 로그 화면을 구현하였다.

4. FastAPI 기반 Backend API 및 WebSocket 구현

- FastAPI 기반 REST API 서버를 구현하였다.
- 문서 업로드, 변환 작업 생성, 작업 상태 조회, Job Item 조회, 문서 목록 조회, 결과 다운로드, 선택 파일 삭제 API를 구현하였다.
- 모델 목록 조회, 대시보드 통계, 시스템 모니터링, 에러 로그, RAG 질의응답 API를 제공하였다.
- CORS 설정과 Nginx 리버스 프록시 구성을 통해 프론트엔드와 백엔드가 /api 경로로 연

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

동되도록 구성하였다.
- WebSocket 기반 진행 이벤트 API를 구현하여 작업의 대기, 시작, 진행, 완료, 실패, 취소 상태를 프론트엔드에 실시간으로 전달하였다.
- Swagger UI를 통해 API 명세를 확인할 수 있도록 구성하였다.

5. RabbitMQ 기반 비동기 작업 처리 구조 구현
- 문서 변환 작업을 API 요청과 분리하기 위해 RabbitMQ 기반 작업 큐 구조를 구현하였다.
- 사용자가 변환 작업을 생성하면 API 서버는 작업 정보를 저장하고 큐에 등록하며, Worker는 큐에서 작업을 가져와 실제 문서 변환을 수행한다.
- 작업 상태는 QUEUED, PROCESSING, COMPLETED, FAILED, CANCELLED 등으로 관리되도록 구현하였다.
- 실패한 작업에 대해서는 최대 재시도 횟수를 적용하여 안정적으로 재처리할 수 있도록 하였다.
- Redis를 활용하여 작업 상태 캐시와 빠른 상태 조회를 지원하고, SQLite를 활용하여 작업 정보, 문서 정보, 변환 결과를 저장하였다.
- OpenAI 기반 worker와 Qwen 로컬 GPU worker 실행 흐름을 분리하여 클라우드 API 기반 처리와 온프레미스 처리를 모두 지원하도록 구성하였다.

6. RAG 기반 문서 검색 및 질의응답 기능 구현
- 변환된 문서를 청크 단위로 분할하고 embedding을 생성하여 문서 검색에 활용하도록 구현하였다.
- 사용자의 질문과 관련성이 높은 문서 내용을 검색한 뒤, 검색 결과를 기반으로 답변을 생성하는 RAG 질의응답 구조를 구현하였다.
- 프론트엔드에는 사용자가 자연어로 문서 내용을 질의할 수 있는 RAG 화면을 제공하였다.
- 표 구조를 단순 텍스트로 평탄화하지 않고 구조화된 단위로 활용할 수 있도록 하여, 문서 기반 검색 정확도를 높이는 방향으로 설계하였다.

7. Docker 기반 배포 및 온프레미스 실행 환경 구축
- Frontend, Backend, Worker, Redis, RabbitMQ를 Docker Compose 기반으로 실행할 수 있도록 구성하였다.
- 일반 실행 환경에서는 OpenAI 또는 OpenRouter API 기반 worker를 활용하고, 온프레미스 환경에서는 Qwen2.5-VL 로컬 GPU worker를 활용할 수 있도록 분리하였다.
- Nginx를 통해 프론트엔드 정적 파일 제공과 /api, /ws 경로 프록시를 지원하도록 구성하였다.
- GHCR(GitHub Container Registry) 기반 이미지 배포와 GitHub Actions 기반 빌드 및 품질 검사 흐름을 구성하였다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

- Docker Compose를 통해 개발 환경과 배포 환경의 실행 방식을 통일하고, 보안이 중요한 내부망 환경에서도 배포 가능한 구조를 마련하였다.

2.2.2 시스템 기능 요구사항

1. 문서 업로드 및 변환

- 드래그 앤 드롭 및 멀티 파일 업로드 지원: 완료
  FileUploader 에서 drag/drop, multiple, 폴더 선택을 지원한다.

- HWP, HWPX, PDF, 이미지 입력 지원: 완료
  백엔드에서 지원 확장자 검증과 문서 처리 파이프라인을 제공한다.

- VLM 모델 선택: 변경
  최초 요구사항은 GPT-5.2, DeepSeek OCR 2 선택이었으나, 실제 구현은 GPT-5.2, GPT-5 Mini, Qwen2.5-vl 7B, OpenRouter Qwen3-vl 32B, Qwen3-피 8B 로 변경되었다.

- 파일 변환은 단일 또는 배치 병렬 처리 가능: 완료
  parallelism, batch API, worker 구조를 통해 단일 및 병렬 변환을 지원한다.

- 변환 결과는 TXT 및 JSON 메타데이터로 제공: 완료
  TXT 출력과 meta/document record 저장 구조가 구현되어 있다.

- 변환 실패 시 재시도 횟수 최대 3 회까지 반영: 완료
  JOB_ITEM_MAX_RETRIES, QWEN_ITEM_MAX_RETRIES 기본값으로 최대 3 회 재시도 구조를 제공한다.

2. 배치 처리 및 모니터링

- 다중 파일 동시 처리: 완료
  병렬 처리 수 입력과 API 반영 구조가 구현되어 있다.

- 실시간 진행률 표시: 변경
  최초 요구사항은 폴링 기반 진행률 표시였으나, 실제 구현은 진행률 API 와 WebSocket 이벤트 기반 실시간 진행 상태 전달 구조로 변경되었다.

- 일시 정지, 재개, 중단 기능: 변경
  중단(cancel)과 재개(resume) 기능은 존재하나, 명확한 일시 정지(pause) 기능은 제한적이다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

- 처리 완료 통계: 완료
  성공, 실패, 소요시간 관련 job/item 상태 및 통계 필드가 존재한다.

3. 파일 관리
- 입력/출력 파일 목록 조회: 완료
  workspace, documents, files 계열 API 를 통해 파일 목록을 조회할 수 있다.

- output 파일 다운로드: 완료
  document download 및 managed download 흐름이 구현되어 있다.

- 선택 파일 삭제: 완료
  document/file delete API 를 통해 선택 파일 삭제가 가능하다.

4. 대시보드
- 전체 작업 현황 확인: 완료
  완료, 진행 중, 대기, 실패 상태를 집계하는 API 가 존재한다.

- 일별 처리량 트렌드 차트: 완료
  프론트엔드에 일별 처리량 차트가 구현되어 있다.

- 성공률 통계: 완료
  프론트엔드에 성공률 통계 차트가 구현되어 있다.

- 시스템 리소스 모니터링: 변경
  CPU load 와 disk 사용량은 구현되어 있으나, 메모리 사용량은 mock 또는 0 값 중심으로 제공된다.

- 최근 작업 내역: 완료
  recent-items API 를 통해 최근 작업 내역을 조회할 수 있다.

5. RAG 질의응답
- 변환된 문서를 벡터 DB 에 로드: 변경
  영구 벡터 DB 가 아니라 OpenAI embedding 과 메모리 기반 SimpleRAG 구조로 구현되었다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

- 채팅 인터페이스를 통한 문서 기반 질의응답: 변경
  채팅 UI 는 존재하나, 실제 RAG 기능은 /rag/query 단발성 API 중심으로 구현되어 있다.

- 참조 문서 표시: 미완료
  답변에 사용된 citations 또는 참조 문서 표시 기능은 완성되지 않았다.

6. 에러 로그
- 변환 실패 내역 조회 및 필터링: 완료
  /monitoring/errors API 를 통해 에러 목록 조회와 필터링이 가능하다.

- 에러 유형 기준 통계: 완료
  /monitoring/errors/summary API 를 통해 에러 유형별 통계를 확인할 수 있다.

- 에러 상세 정보 확인: 완료
  /monitoring/errors/{errorId} API 를 통해 에러 상세 정보를 확인할 수 있다.

2.2.3 시스템 비기능(품질) 요구사항

1. 보안
- 사용자의 API 키는 암호화되어 저장되지 않고 세션에서만 유지: 변경
  사용자별 API 키를 세션에 저장하는 방식이 아니라, 서버 환경변수 OPENAI_API_KEY 를 중심으로 관리하는 방식으로 변경되었다.

- 사용자의 로그인은 JWT Token 을 이용하여 Stateless 하게 유지: 미달성
  JWT 기반 stateless 인증이 아니라 AUTH_SESSIONS 에 저장되는 Bearer 세션 토큰 방식으로 구현되었다.

- VLM 모델 선택: 변경
  DeepSeek OCR 2 는 제외되고 GPT, OpenRouter, Qwen 중심의 모델 선택 구조로 변경되었다.

- 파일 업로드 크기 제한 및 확장자 검증: 변경
  HWP, HWPX, PDF, 이미지, Excel 등 지원 파일 형식을 판별하고 처리 가능 여부를 표시하는 확장자 검증은 구현되었으나, 명확한 업로드 크기 제한은 확인되지 않았다.

- 입출력 디렉토리 격리: 달성

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

resolve_managed_path 에서 input, output, tmp root 하위 여부를 검증하여 path traversal 을 방지한다.

2. 성능

- 단일 파일 변환, HWP 30 페이지 기준 3.5 분 이내: 달성

- 배치 처리, 병렬 4 기준 10 개 파일 10 분 이내: 달성

- 프론트엔드 초기 로딩 3 초 이내: 달성

3. 확장성

- 새로운 VLM 모델 추가 용이: 달성
  VLMClient, 모델 카탈로그, 실행 백엔드 라우팅 구조가 존재한다.

- 새로운 문서 형식 지원 추가 용이: 달성
  converter 와 pipeline 모듈화 구조를 통해 문서 형식 확장이 가능하다.

- Docker
  기반 수평 확장 가능: 달성
  Worker 분리, RabbitMQ/Redis 기반 확장 구조를 통해 수평 확장 기반을 제공한다.

4. 사용성

- 직관적인 드래그 앤 드롭 파일 업로드: 달성
  드래그 앤 드롭 기반 업로드 기능이 구현되어 있다.

- 실시간 진행 상태 확인: 달성
  진행률 API 와 WebSocket 구조를 통해 실시간 상태 확인이 가능하다.

- 반응형 웹 디자인: 달성
  Tailwind 기반 반응형 UI 구조가 적용되어 있다.

5. 유지보수성

- FSD 아키텍처 기반 프론트엔드 레이어 분리: 달성
  app, pages, widgets, features, entities, shared 구조로 레이어가 분리되어 있다.

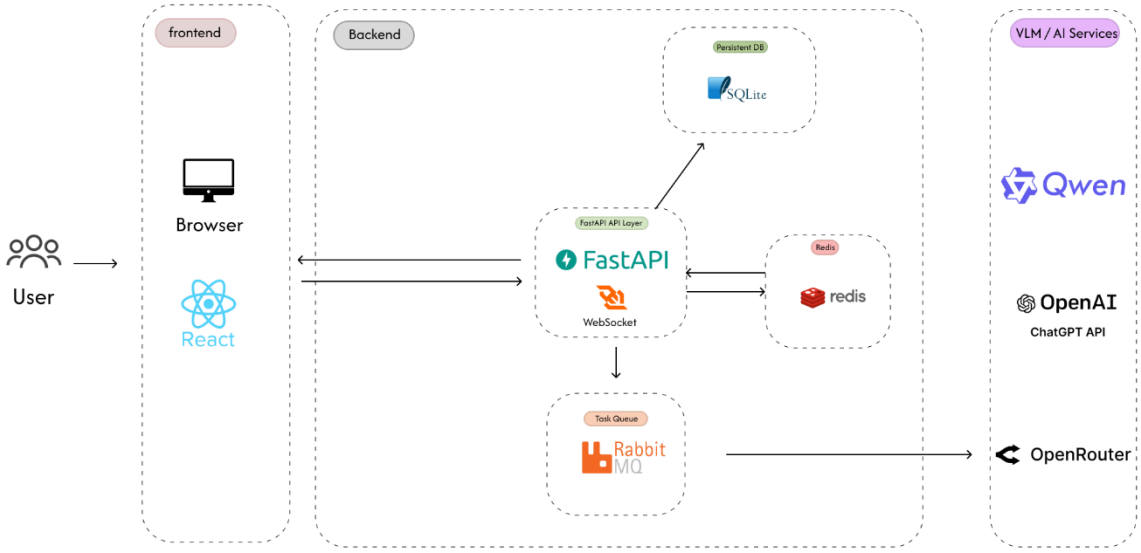
All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

- ESLint + Prettier + Husky 코드 품질 관리: 달성
  package.json 에 ESLint, Prettier, Husky 설정이 구성되어 있다.

- Storybook 컴포넌트 문서화: 달성
Storybook 스크립트와 설정이 존재한다.

2.2.1 시스템 구조 및 설계도

시스템 아키텍처



1. Backend 와 Frontend 를 분리한 이유

- 화면 개발과 서버 로직을 독립적으로 개발할 수 있기 때문

- API 기반 구조라서 기능 확장이 용이하다.

-문서 변환처럼 서버 자원을 많이 사용하는 작업을 브라우저가 직접 처리하지 않고, 백엔드와 Worker 가 담당하도록 하기 위함이다..

- 추후 배포 시 fe/be 를 각각 확장 가능

2. Fast API 를 API Layer 로 둔 이유

- 문서 업로드, 변환요청, 작업 상태 조회, 파일 다운로드, RAG 질의응답 같은 기능들은 모두 API 로 처리됨

- 파일 업로드 API

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

- 변환 작업 생성 API
- 작업 상태 조회 API
- 결과 문서 조회/다운로드 API
- 에러 로그 및 대시보드 API
- WebSocket 기반 진행률 알림

3. Fast API 를 선택이유

-Fast API 는 Python 기반이라 문서 처리 라이브러리와 OpenAI API, Qwen 과 같은 AI 모델 연동과 잘 맞기 때문이다.

4. WebSocket 을 둔 이유

- 문서 변환은 즉시 끝나는 작업이 아니라 몇 초에서 몇 분까지 걸릴 수 있기 때문에, 사용자가 변환 중인지 몇 번째 파일을 처리중인지 또는 실패했는지 알 수 있어야 하기 때문에 실시간 진행률 전달이 필요했음

- 따라서 FastAPI 안에 WebSocket 기능을 두고, 워커나 job 상태가 바뀔 때 프론트엔드로 진행 이벤트를 전달하는 구조가 되었음

- 사용자가 화면을 보면서 작업 흐름을 확인 할 수 있다.

5. SQLite 를 Persistent DB 로 둔 이유

- 작업 상태, 문서 정보, 사용자 정보, 결과 메타데이터는 서버가 재시작 되어도 남아 있어야 하기 때문에 영구 저장소가 필요했기 때문

-별도 DB 서버 설치 없이 바로 사용 가능

- 로컬 개발과 테스트가 쉬움

- 문서/작업 메타데이터 저장에 충분함

6. Redis 를 둔이유

- Redis 는 빠르게 변하는 상태 정보를 캐싱하거나, 작업 상태를 빠르게 조회하기 위해 사용됨

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

-문서 변환 작업은 대기 중, 처리 중, 완료, 실패, 취소 요청, 진행률 갱신 등 상태가 자주 바뀌고, 이러한 데이터는 빠르게 읽고 쓸 필요가 있기 때문이다.

7. RabbitMQ 를 Task Queue 로 둔 이유

-문서 변환과 AI 분석은 시간이 오래걸리고 서버 자원을 많이 사용함. 따라서 변환 요청은 RabbitMQ 에 작업으로 넣고, worker 가 큐에서 하나씩 꺼내 처리하는 구조

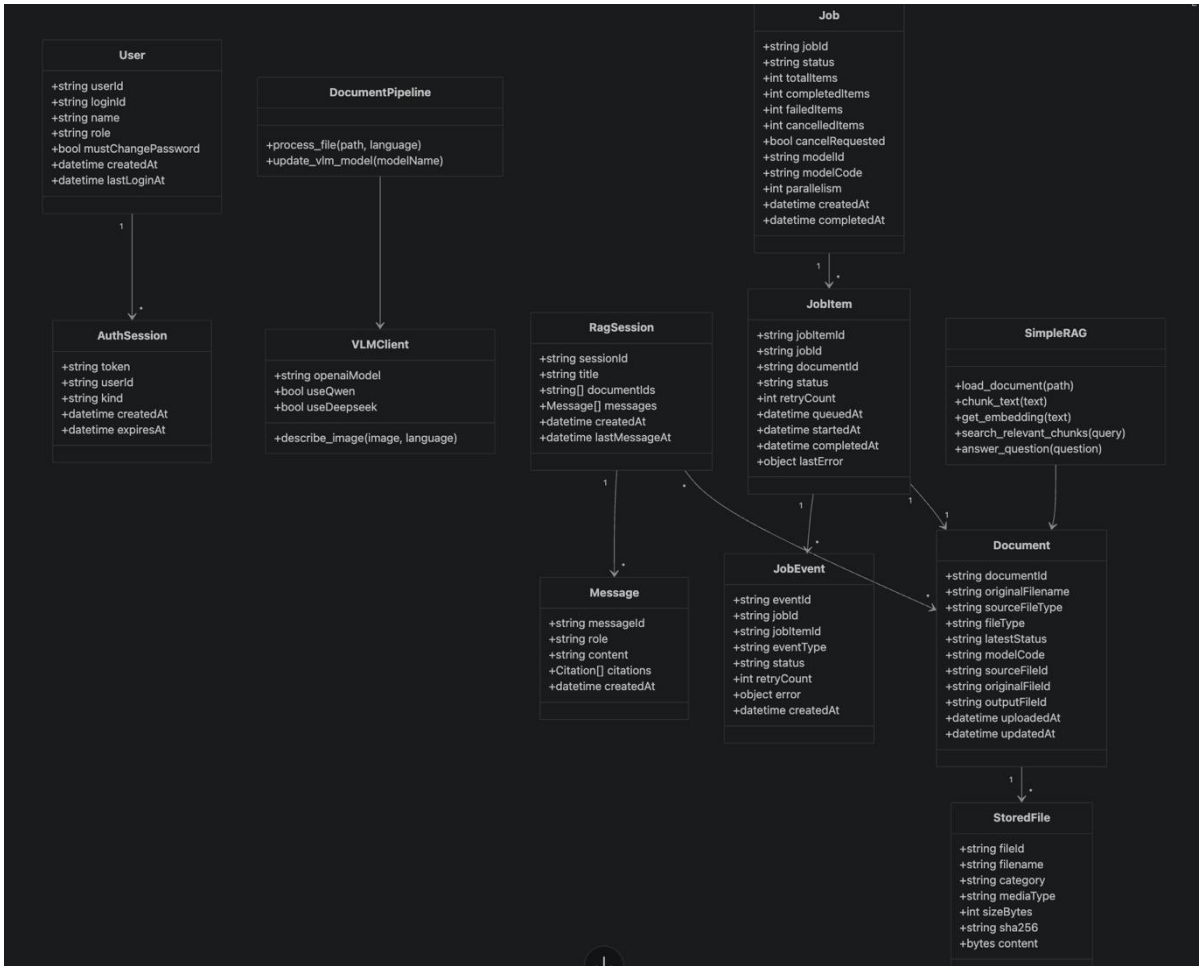
- 오래 걸리는 변환 작업을 API 요청과 분리
- 여러 worker 로 병렬 처리 가능
- 실패 시 재시도 구조 만들기 쉬움

-OpenAI worker, Qwen worker 처럼 작업 종류별 분리 가능

- 대량 파일 배치 처리에 적합

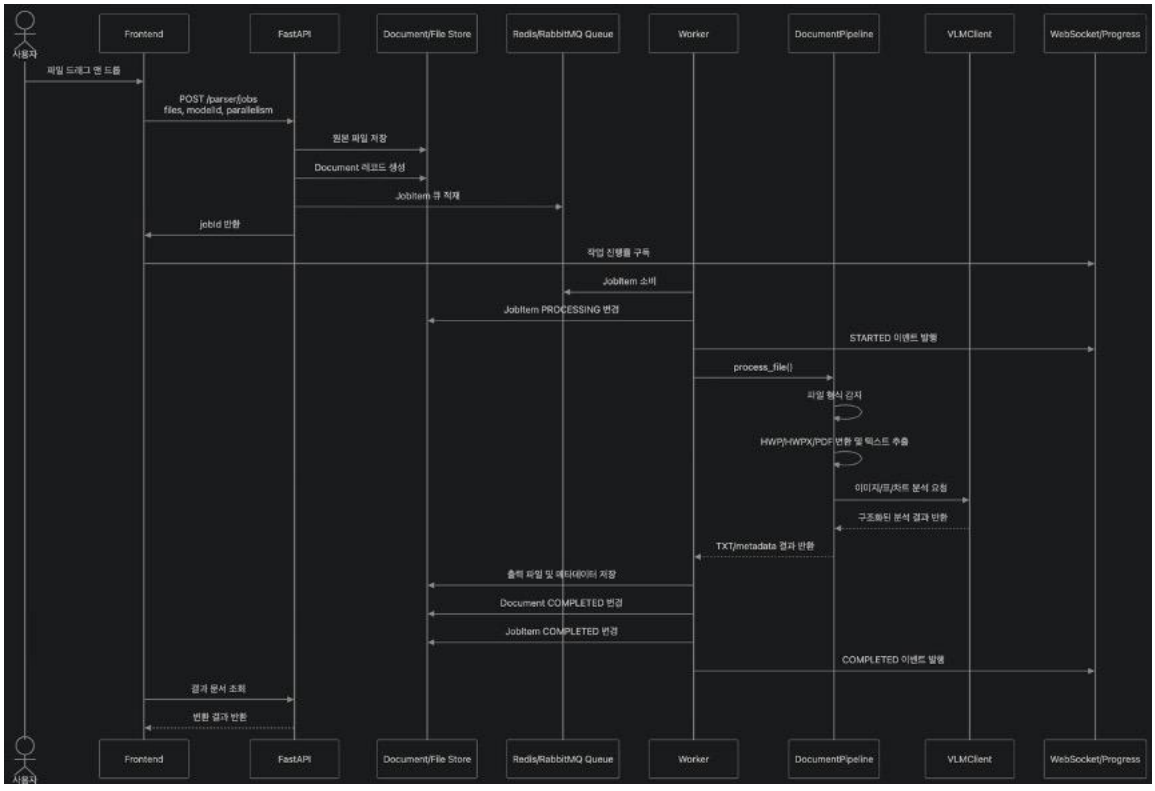
All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

클래스 다이어그램



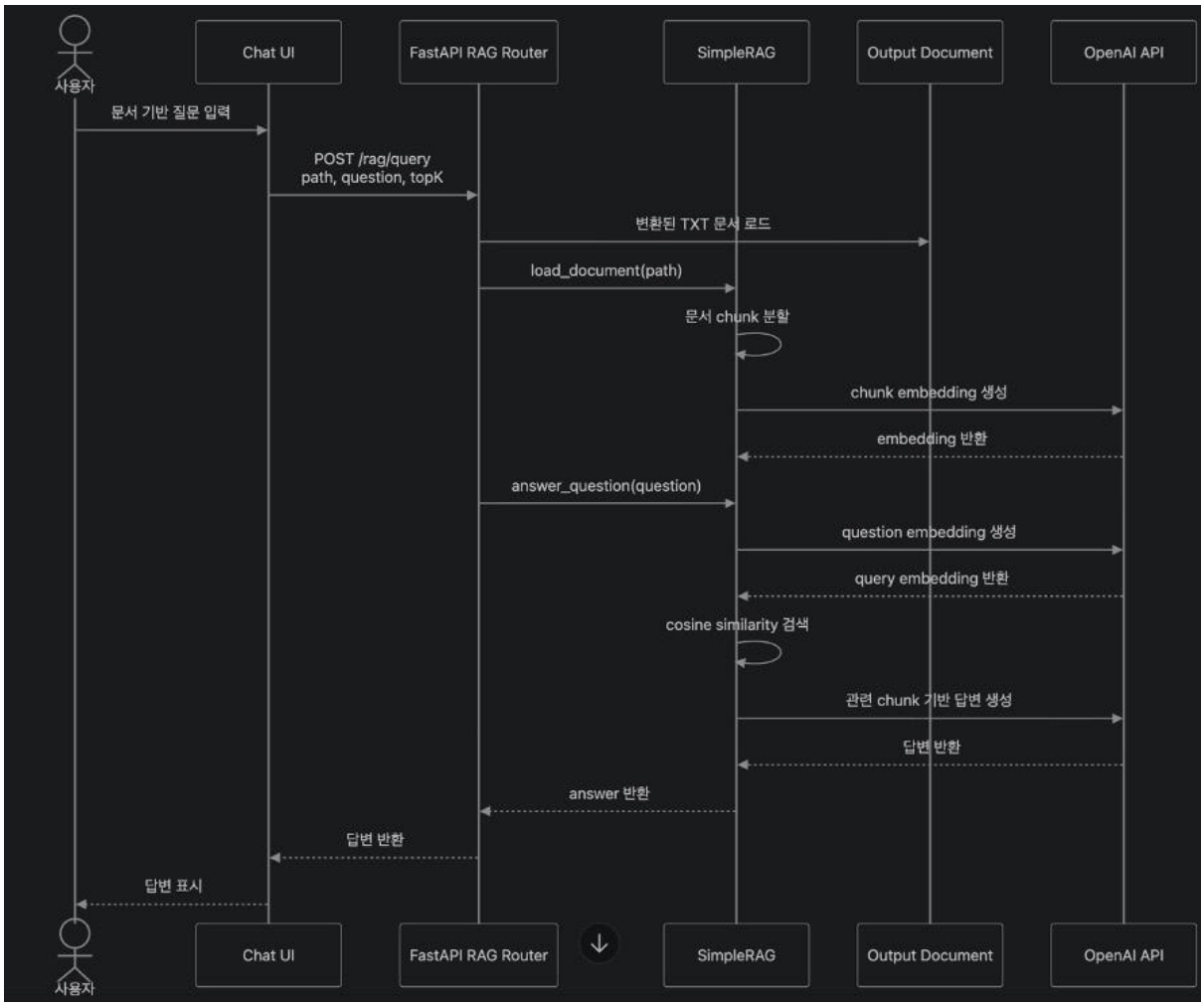
All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

시퀀스 다이어그램 : 문서 업로드 및 변환



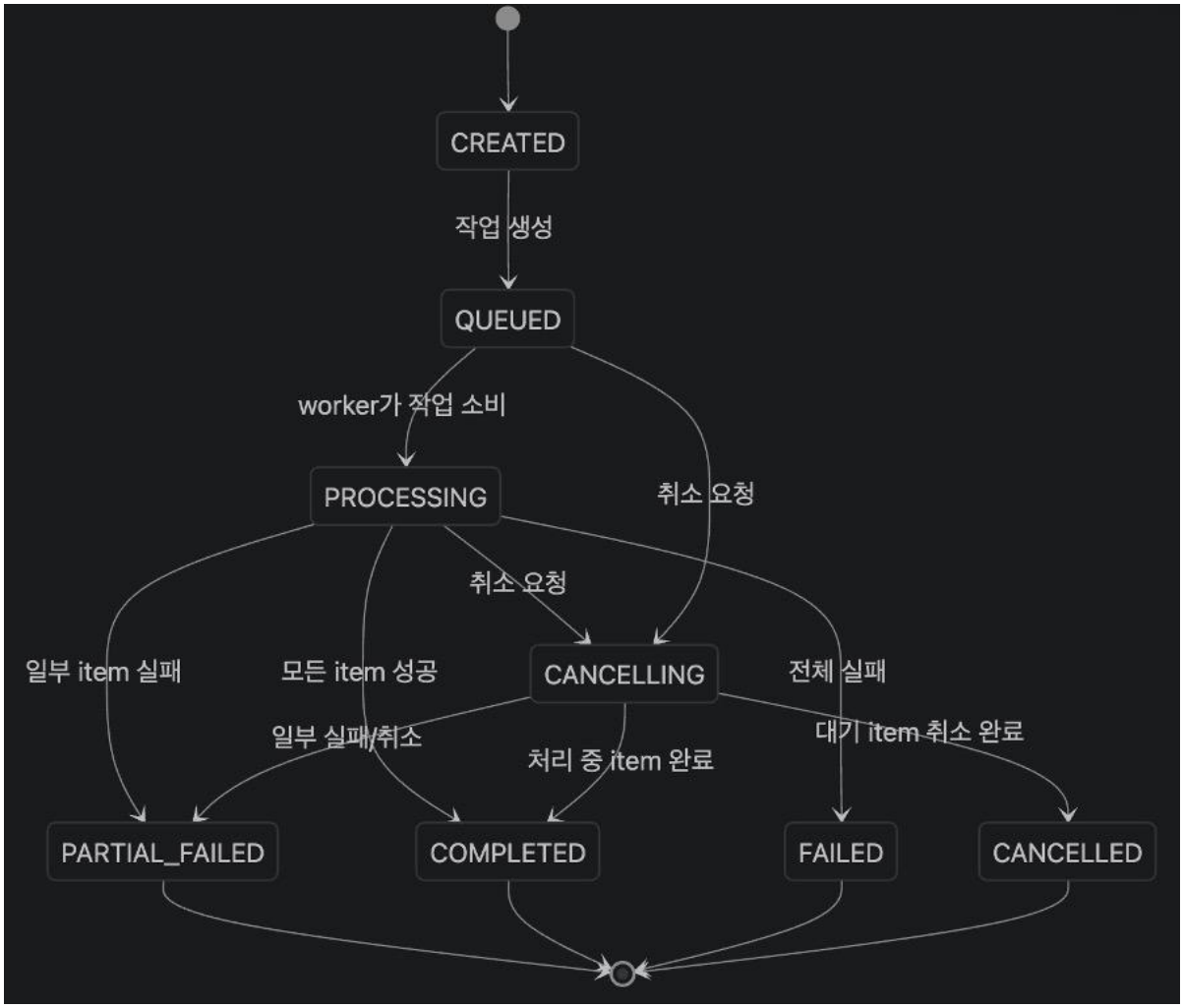
All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

시퀀스 다이어그램 : RAG 질의응답



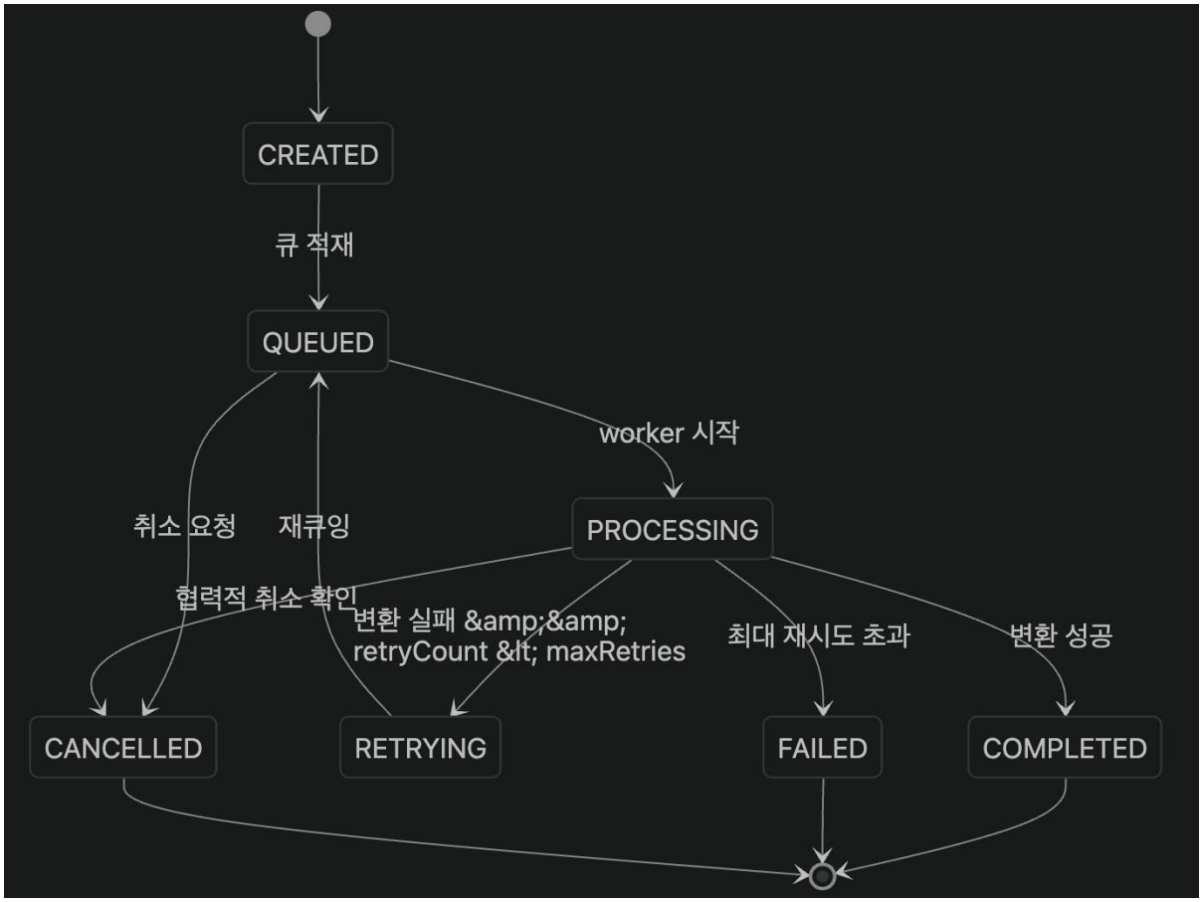
All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

상태 다이어그램 : job 상태



All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

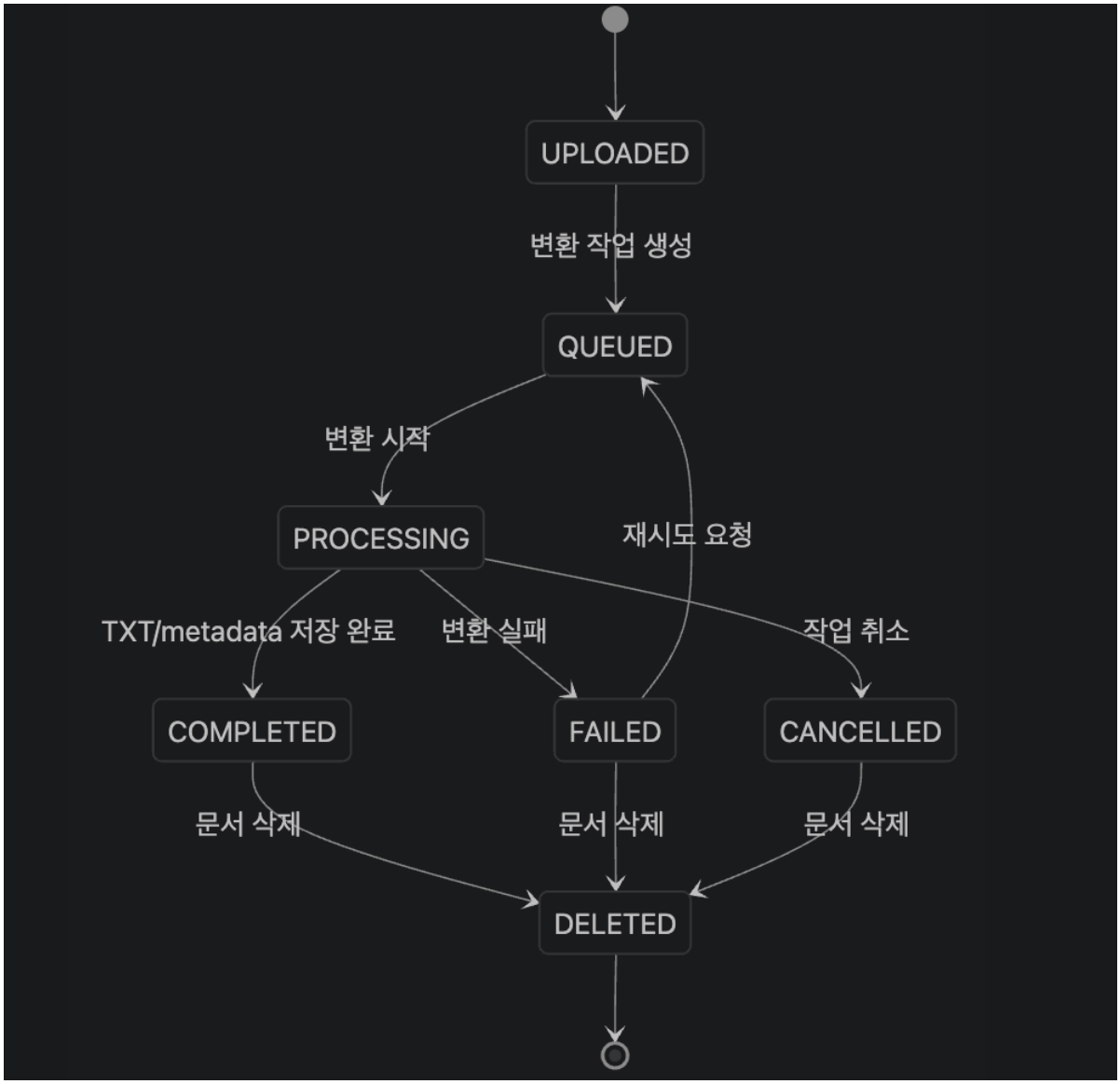
상태 다이어그램 : jobitem 상태



All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

상태 다이어그램 : Doucment 상태

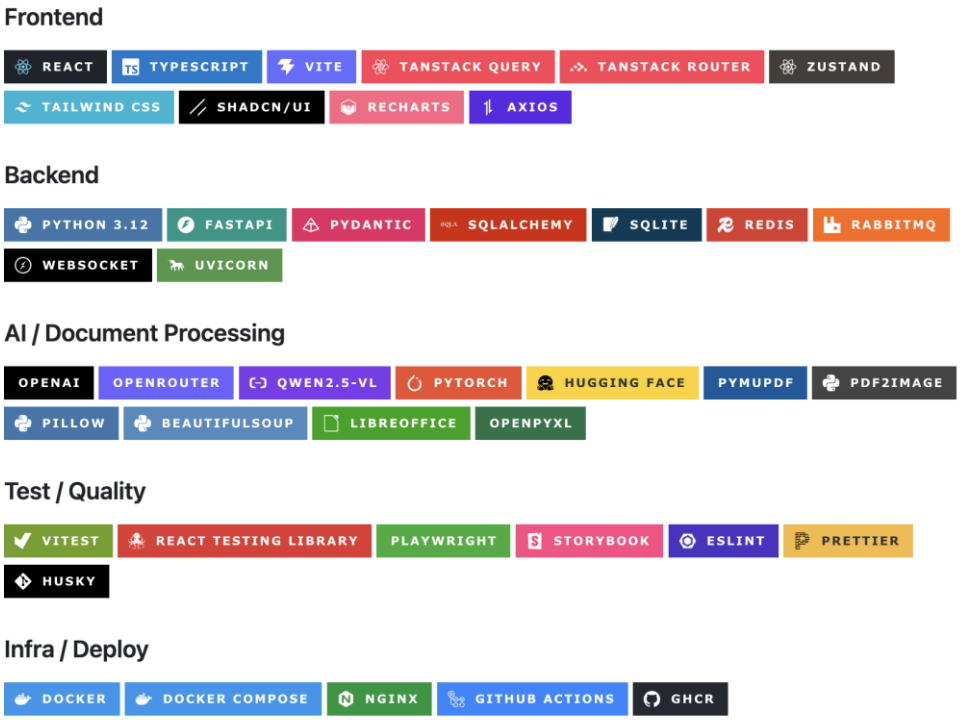
All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.



All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

2.2.2 활용/개발된 기술

1. 개발에 활용된 기술



2.개발된 기술

AI 기반 문서 파싱 기술

- HWP, HWPX, PDF, 이미지 파일을 입력받아 텍스트, 표, 이미지 정보를 추출하는 문서 분석 파이프라인을 개발했다.

- 일반 텍스트와 네이티브 표 구조는 문서 파서로 우선 추출하고, 일반 파싱만으로 처리하기 어려운 표, 차트, 이미지 영역은 VLM 모델을 활용해 HTML/Markdown 등 구조화된 결과로 변환할 수 있다.

HWP/HWPX/PDF 변환 및 구조화 기술

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

- HWP 문서는 텍스트 직접 추출 경로를 우선 적용하고, 직접 추출이 어려운 경우 PDF 변환 기반 fallback을 통해 페이지 단위 분석을 수행하도록 구현했다.

- HWPX 문서는 XML 구조를 직접 파싱하여 본문 텍스트, 표, 이미지 정보를 추출하고, 내장 이미지 영역은 VLM 모델을 활용해 구조화된 설명 결과로 보강하도록 구현했다.

- PDF 문서는 페이지 단위로 텍스트, 표, 이미지 블록을 추출하고, 일반 파싱만으로 처리하기 어려운 이미지성 표·차트·흐름도 영역은 VLM 기반 분석을 통해 HTML/Markdown 등 구조화된 결과로 변환하도록 구현했다.

비동기 문서 변환 작업 처리 기술

- 문서 변환 작업을 API 요청과 분리하기 위해 RabbitMQ 기반 작업 큐 구조를 개발했다. 작업 상태를 QUEUED, PROCESSING, COMPLETED, FAILED, CANCELLED 등으로 관리하고, 실패 시 재시도할 수 있도록 구현했다.

실시간 진행률 모니터링 기술

- WebSocket을 활용해 문서 변환 진행 상황을 프론트엔드에 실시간으로 전달한다. 사용자는 업로드한 파일의 변환 시작, 진행, 완료, 실패 상태를 화면에서 확인할 수 있다.

AI 모델 라우팅 기술

- GPT-5.2, GPT-5 Mini, Qwen2.5-VL 7B, Qwen3-VL 8B, Qwen3-VL 32B 등 여러 VLM 모델을 선택적으로 사용할 수 있도록 모델 카탈로그와 실행 백엔드 라우팅 구조를 개발했다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

- OpenAI API, OpenRouter 기반 클라우드 모델과 Qwen2.5-VL 7B 로컬 GPU 모델을 분리하여, 배포 환경과 모델 종류에 따라 다른 실행 방식을 사용할 수 있도록 구현했다.

RAG 기반 문서 질의응답 기술

- 변환된 문서를 청크 단위로 나누고 embedding을 생성해, 사용자의 질문과 유사한 문서 내용을 검색하는 구조를 구현했다.
- 검색된 문서 내용을 기반으로 답변을 생성하는 RAG 질의응답 기능을 개발했다.

문서 관리 및 대시보드 기술

- 변환된 문서 목록 조회, 다운로드, 삭제 기능을 개발했다.
- 전체 작업 현황, 성공률, 일별 처리량, 최근 작업 내역, 에러 로그를 확인할 수 있는 대시보드 기능을 구현했다.

2.2.3 현실적 제한 요소 및 그 해결 방안

1. VLM 분석의 GPU 의존성

문서 내 표, 이미지, 차트, 수식과 같은 복합 요소를 정밀하게 분석하기 위해서는 VLM 기반 추론이 필요하다. 특히 Qwen2.5-VL 계열 로컬 모델을 사용할 경우 GPU 자원이 필요하며, 모델 크기와 처리량에 따라 높은 VRAM이 요구된다.

이를 해결하기 위해 본 프로젝트는 클라우드 API 기반 모델과 로컬 GPU 모델을 모두 지원하는 이중 모델 구조를 채택하였다. GPU가 없는 환경에서는 OpenAI GPT 계열 또는 OpenRouter 기반 모델을 사용하여 클라우드 방식으로 문서를 처리할 수 있고, GPU가 있는 온프레미스 환경에서는 Qwen2.5-VL 모델을 사용하여 로컬 추론이 가능하도록 구성하였다. 또한 GPU 환경에서는 동시 추론 수와 worker 병렬성을 제한할 수 있도록 하여 GPU 메모리 부족 문제를 완화하였다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

2. 대용량 문서 처리 시 메모리 사용량 증가

수십~수백 페이지의 HWP, PDF 문서를 처리할 경우 전체 문서를 한 번에 메모리에 올리면 메모리 사용량이 급격히 증가할 수 있다. 특히 PDF 페이지 렌더링, 이미지 추출, VLM 입력 이미지 생성 과정에서 메모리 사용량이 커질 수 있다.

이를 해결하기 위해 문서를 페이지 단위로 처리하는 구조를 적용하였다. 각 페이지에서 텍스트 블록과 이미지 영역을 추출하고, 분석이 끝난 중간 객체는 즉시 해제하는 방식으로 메모리 사용량을 줄였다. 또한 변환 작업을 Worker 단위로 분리하고, 병렬 처리 수를 설정할 수 있도록 하여 서버 사양에 맞게 처리량과 메모리 사용량을 조절할 수 있도록 하였다.

3. HWP 변환의 Linux 환경 의존성

HWP/HWPX 문서 변환에는 hwp5html, LibreOffice, wkhtmltopdf 등 외부 변환 도구가 필요하다. 이러한 도구들은 운영체제와 설치 환경에 따라 동작 방식이 달라질 수 있으며, 특히 Windows와 macOS 개발 환경에서는 동일한 결과를 보장하기 어렵다.

이를 해결하기 위해 Docker 기반 실행 환경을 제공하였다. Docker 컨테이너 내부에 필요한 문서 변환 도구와 Python 의존성을 포함하여, 개발자 환경과 운영 환경의 차이를 줄였다. 이를 통해 HWP 변환 과정에서 발생할 수 있는 환경별 오류를 최소화하고, Linux 기반의 일관된 실행 환경에서 문서 처리가 가능하도록 하였다.

4. HWP → PDF 변환 실패 가능성

일부 HWP 파일은 변환 과정에서 원본 구조가 깨지거나, OLE 바이너리 데이터가 비정상적으로 렌더링되어 수백~수천 페이지의 잘못된 PDF가 생성되는 문제가 발생할 수 있다. 또한 표나 이미지가 포함된 문서는 단일 변환 방식만으로 안정적인 결과를 얻기 어렵다.

이를 해결하기 위해 3단계 폴백 전략을 적용하였다. 우선 hwp5html 기반 변환을 시도하고, 실패하거나 결과가 불완전한 경우 LibreOffice 기반 변환을 수행한다. 이마저 실패할 경우 텍스트 기반 PDF 생성 방식으로 전환하여 최소한의 텍스트 추출 결과를 확보한다. 또한 변환된 PDF의 페이지 수와 내용 품질을 검증하고, OLE 마커나 비정상 페이지 패턴을 감지하여 쓰레기 페이지를 자동으로 걸러내는 안정성 보완 로직을 적용하였다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

5. OpenAI API 비용 증가

OpenAI 또는 OpenRouter 기반 VLM API를 사용할 경우, 문서 페이지 수와 이미지 분석 요청 수가 증가할수록 API 호출 비용이 커진다. 특히 대량 문서 변환이나 반복적인 배치 처리 환경에서는 비용 부담이 주요 제한 요소가 된다.

이를 해결하기 위해 로컬 GPU 기반 Qwen2.5-VL 모델 실행 구조를 함께 제공하였다. 클라우드 API 사용이 적합한 환경에서는 OpenAI 또는 OpenRouter를 활용하고, 대량 문서를 반복적으로 처리해야 하는 환경에서는 Qwen2.5-VL 로컬 모델을 활용할 수 있도록 하였다. 이를 통해 초기 GPU 자원은 필요하지만, 장기적으로 대량 문서 처리 비용을 절감할 수 있는 선택지를 제공하였다.

6. 폐쇄망 및 보안 환경 지원

공공기관, 금융기관, 연구기관과 같이 보안이 중요한 환경에서는 문서 원본을 외부 클라우드 API로 전송하기 어렵다. 따라서 OpenAI API에만 의존하는 구조는 폐쇄망 또는 내부망 환경에서 활용하기 어렵다는 한계가 있다.

이를 해결하기 위해 온프레미스 실행 구조를 지원하였다. Qwen2.5-VL 로컬 GPU 모델을 사용하면 문서를 외부 서버로 전송하지 않고 내부 서버에서 분석할 수 있다. 또한 Docker Compose 기반으로 Backend, Worker, Redis, RabbitMQ, Qwen worker를 내부망에 배포할 수 있도록 구성하여, 보안 요구가 높은 환경에서도 문서 파싱 기능을 사용할 수 있도록 하였다.

7. 장시간 변환 작업의 사용자 경험 문제

문서 변환은 파일 크기, 페이지 수, AI 분석 여부에 따라 시간이 오래 걸릴 수 있다. 사용자가 작업 진행 상태를 확인할 수 없다면 변환이 정상적으로 진행 중인지, 실패했는지 알기 어렵다.

이를 해결하기 위해 WebSocket 기반 실시간 진행 상태 전달 기능을 구현하였다. 작업의 대기, 시작, 진행, 완료, 실패, 취소 상태를 프론트엔드에 실시간으로 전달하여 사용자가 진행률을 즉시 확인할 수 있도록 하였다. 또한 작업 상태를 저장소에 기록하여 새로고침이나 재접속 후에도 현재 작업 상태를 확인할 수 있도록 하였다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

8. 비동기 작업 처리 및 장애 복구

문서 변환 작업을 API 요청 내에서 직접 처리하면 요청 시간이 길어지고, 서버 장애나 네트워크 오류가 발생했을 때 작업 상태를 추적하기 어렵다. 또한 여러 사용자가 동시에 문서를 업로드할 경우 API 서버에 부하가 집중될 수 있다.
이를 해결하기 위해 본 프로젝트는 문서 변환 작업을 API 요청과 분리하고, RabbitMQ 기반 작업 큐를 통해 Worker가 비동기적으로 처리하도록 구현하였다. 사용자가 변환 작업을 요청하면 API 서버는 작업 정보를 저장한 뒤 큐에 등록하고, Worker는 큐에서 작업을 가져와 실제 문서 변환을 수행한다. 작업 상태는 SQLite와 Redis를 통해 관리되며, 실패한 작업은 재시도 정책에 따라 다시 처리할 수 있도록 구성하였다. 또한 WebSocket을 통해 작업 진행 상태를 프론트엔드에 전달하여, 사용자가 장애 또는 실패 상황을 즉시 확인할 수 있도록 하였다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

2.2.4 결과물 목록

번호	프로젝트 결과물	주요 내용	기술문서 유/무	부록 삽입 문서
1	AI 기반 문서 파싱 시스템	HWP, HWPX, PDF, 이미지 파일을 텍스트, 표, 이미지 정보로 변환	유	시스템 아키텍처 문서
2	문서 변환 파이프라인	파일 형식 감지, PDF 변환, 페이지 단위 추출, VLM 기반 분석, 결과 생성	유	문서 처리 파이프라인 설명서
3	Backend API 서버	문서 업로드, 변환 작업 생성, 상태 조회, 다운로드, 삭제, RAG API 제공	유	API 명세서, Backend README
4	Frontend 웹 애플리케이션	문서 업로드, 변환 설정, 진행 상태 확인, 결과 미리보기, 대시보드 제공	유	Frontend 프로젝트 가이드
5	비동기 작업 처리 시스템	RabbitMQ 기반 작업 큐, Worker 처리, 작업 재시도 및 취소 기능 제공	유	Queue API 문서, Worker Routing 문서

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

6	실시간 진행률 모니터링	WebSocket 기반 작업 시작, 진행, 완료, 실패 이벤트 전달	유	API 명세서
7	문서 및 결과 파일 관리 기능	입력/출력 파일 목록 조회, 미리보기, 다운로드, 삭제 기능 제공	유	사용자 매뉴얼
8	대시보드 기능	전체 작업 현황, 성공률, 일별 처리량, 최근 작업 내역 확인	유	사용자 매뉴얼
9	에러 로그 관리 기능	변환 실패 내역 조회, 에러 유형별 통계, 에러 상세 정보 확인	유	운영자 매뉴얼
10	RAG 질의응답 기능	변환된 문서를 기반으로 문서 검색 및 질의응답 수행	유	사용자 매뉴얼, API 명세서
11	Docker 기반 배포 환경	Frontend, Backend, Worker, Redis, RabbitMQ를 Docker Compose로 통합 실행	유	배포 가이드
12	온프레미스 Qwen 실행 환경	Qwen2.5-VL 로컬 GPU worker를 활용한 내부망 문서 분석 지원	유	Qwen 실행 가이드, 배포 가이드

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

2.3 기대효과 및 활용방안

1. 한국형 문서 데이터 인프라 구축

본 프로젝트는 HWP/HWPX, PDF, 이미지, Excel 등 국내 업무 환경에서 널리 사용되는 문서를 AI가 활용 가능한 구조화 데이터로 변환하는 기반을 구축한다. 기존 글로벌 문서 처리 도구가 충분히 지원하지 못하는 HWP 중심 문서 환경을 보완하고, 공공·민간에 축적된 비정형 문서를 검색, 질의응답, 분석에 활용 가능한 데이터 자원으로 전환한다. 이를 통해 한국형 문서에 특화된 AI 데이터 파이프라인을 확보할 수 있다.

2. 공공·기업 업무 자동화 및 지식관리 고도화

본 시스템은 공공기관의 행정문서, 기업의 보고서·계약서·내부 규정, 연구기관의 연구자료와 같은 문서를 자동으로 분석하고 구조화하여 업무 효율을 높일 수 있다. 문서 검토, 정보 추출, 정리, 검색에 소요되는 반복 작업을 줄이고, 조직 내 문서 자산을 지식관리 시스템(KMS), 그룹웨어, 업무 자동화 시스템과 연계 가능한 데이터로 전환할 수 있다. 이를 통해 공공 행정과 민간 업무 모두에서 문서 기반 의사결정과 업무 자동화를 지원한다.

3. RAG 기반 문서 검색 및 질의응답 서비스 확장

변환된 문서는 Markdown, HTML table, JSON 메타데이터, 일반 텍스트 형태로 제공되므로 RAG 기반 검색 및 질의응답 시스템에 바로 활용할 수 있다. 사용자는 문서 내용을 직접 열람하지 않아도 질문을 통해 필요한 정보를 찾을 수 있으며, 내부 문서 기반 AI 챗봇, 업무 매뉴얼 검색, 계약서 검토, 행정문서 검색 서비스 등으로 확장할 수 있다. 이는 기존 단순 OCR 결과보다 문서의 구조와 의미를 보존한다는 점에서 검색 품질 향상에 기여한다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

"궁극적 목표 : 공공, 기업 문서의 안전한 AI 활용 촉진"

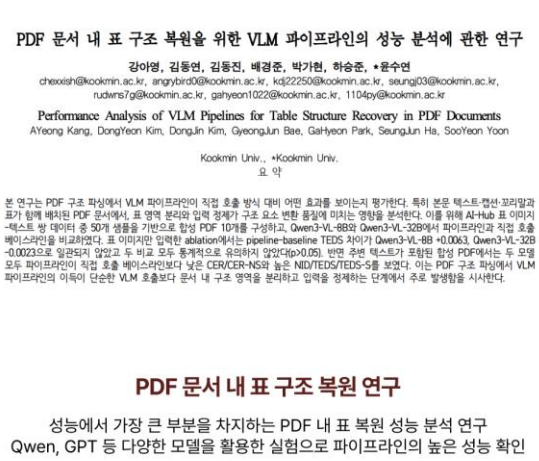
4. 온프레미스 기반 보안형 문서 AI 활용

공공기관, 금융기관, 연구기관 등은 보안 정책상 문서를 외부 클라우드 API로 전송하기 어렵다. 본 프로젝트는 OpenAI/OpenRouter 기반 외부 API 처리뿐만 아니라 Qwen2.5-VL 기반 로컬 GPU 추론 구조를 고려하여, 내부망 또는 폐쇄망에서도 문서 분석이 가능하도록 한다. 이를 통해 민감한 문서를 외부로 전송하지 않고도 AI 기반 문서 파싱과 RAG 검색을 수행할 수 있으며, 보안 요구가 높은 기관에서도 활용 가능성을 확보한다.

5. 기술 검증 및 연구 성과 확보

본 프로젝트는 단순 구현에 그치지 않고, 문서 파싱 성능과 병렬 처리 성능을 분석하는 연구로 확장되었다. PDF 문서 내 표 구조 복원을 위한 VLM 파이프라인의 성능을 분석하고, RAG 적용을 위한 VLM 기반 문서 파싱 파이프라인의 병렬 처리 성능을 비교하였다. 이를 통해 Qwen, GPT 등 다양한 모델을 활용한 표 구조 복원 성능과 순차 처리 대비 병렬 처리의 속도 향상 효과를 검증하였으며, 프로젝트 결과를 논문 형태로 정리하여 기술적 타당성과 연구 성과를 확보하였다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.



PDF 문서 내 표 구조 복원을 위한 VLM 파이프라인의 성능 분석에 관한 연구

강아영, 김동연, 김동진, 배경준, 박가현, 하승준, *윤수연

chexxish@kookmin.ac.kr, angrybird0@kookmin.ac.kr, kdj22250@kookmin.ac.kr, seungj03@kookmin.ac.kr, rudwns7g@kookmin.ac.kr, gahyeon1022@kookmin.ac.kr, 1104py@kookmin.ac.kr

Performance Analysis of VLM Pipelines for Table Structure Recovery in PDF Documents

AYeong Kang, DongYeon Kim, DongJin Kim, GyeongJun Bae, GaHyeon Park, SeungJun Ha, SooYeon Yoon

Kookmin Univ., *Kookmin Univ.

요 약

본 연구는 PDF 구조 파싱에서 VLM 파이프라인이 직접 호출 방식 대비 어떤 효과를 보이는지 평가한다. 특히 본문 텍스트·캡션·꼬리말과 표가 함께 배치된 PDF 문서에서, 표 영역 분리와 입력 정제가 구조 요소 변환 품질에 미치는 영향을 분석한다. 이를 위해 AI-Hub 표 이미지 -텍스트 쌍 데이터 중 50개 샘플을 기반으로 합성 PDF 10개를 구성하고, Qwen3-VL-8B와 Qwen3-VL-32B에서 파이프라인과 직접 호출 베이스라인을 비교하였다. 표 이미지만 입력한 ablation에서는 pipeline-baseline TEDS 차이가 Qwen3-VL-8B +0.0063, Qwen3-VL-32B -0.0023으로 일관되지 않았고 두 비교 모두 통계적으로 유의하지 않았다(p>0.05). 반면 주변 텍스트가 포함된 합성 PDF에서는 두 모델 모두 파이프라인이 직접 호출 베이스라인보다 낮은 CER/CER-NS와 높은 NID/TEDS/TEDS-S를 보였다. 이는 PDF 구조 파싱에서 VLM 파이프라인의 이득이 단순한 VLM 호출보다 문서 내 구조 영역을 분리하고 입력을 정제하는 단계에서 주로 발생함을 시사한다.

RAG 적용을 위한 VLM 기반 문서 파싱 파이프라인의 병렬처리 성능 분석에 관한 연구

배경준, 강아영, 김동연, 김동진, 박가현, 하승준, *윤수연

rudwns7g@kookmin.ac.kr, chexxish@kookmin.ac.kr, angrybird0@kookmin.ac.kr, kdj22250@kookmin.ac.kr, gahyeon1022@kookmin.ac.kr, seungj03@kookmin.ac.kr, 1104py@kookmin.ac.kr

A Study on the Parallel Processing Performance Analysis of a VLM-Based Document Parsing Pipeline for RAG Applications

GyeongJun Bae, AYeong Kang, DongYeon Kim, DongJin Kim, GaHyeon Park, SeungJun Ha, *SooYeon Yoon

Kookmin Univ., *Kookmin Univ.

요 약

본 논문은 RAG 적용을 위한 문서 데이터 생성을 목적으로, 본 연구에서 구현한 시각언어모델(Vision-Language Model, VLM) 기반 문서 파서의 파이프라인과 병렬처리 성능을 분석한다. 본 시스템은 HWP, HWPX, PDF, IMG 등의 문서를 입력으로 받아 표, 차트, 흐름도와 같은 시각 요소를 구조화된 텍스트로 변환하여 RAG 기반 질의응답에 활용 가능한 문서 데이터를 생성한다. 단순 텍스트 추출 방식은 문서 내 표, 차트, 흐름도 등의 구조를 보존하지 못한 채 추출하는 문제가 있고, 순차 처리 방식은 이미지 추출, 문서 변환, VLM 추론, 결과 병합의 과정이 차례대로 수행되어 CPU를 사용하는 전·후처리에서 GPU가 유휴 상태가 되거나, VLM 추론 중 CPU 기반 작업이 대기하는 문제가 있다. 이를 완화하기 위해 본 연구

PDF 문서 내 표 구조 복원 연구

성능에서 가장 큰 부분을 차지하는 PDF 내 표 복원 성능 분석 연구
Qwen, GPT 등 다양한 모델을 활용한 실험으로 파이프라인의 높은 성능 확인

파이프라인의 병렬처리 성능 분석 연구

효율적인 비동기 문서처리 구조, 변환 방식 연구
순차 처리 방식 대비 속도 향상, 파이프라인을 통한 RAG 정확도 향상 확인

6. 문서 AI 플랫폼으로의 확장 가능성

본 시스템은 문서 파싱 기능을 중심으로 시작하지만, 향후 기업 지식관리, 행정문서 자동처리, 컴플라이언스 문서 분석, 계약서 검토, 전자결재 연동, ERP·그룹웨어 연동 등 다양한 문서 AI 플랫폼으로 확장될 수 있다. API 기반 구조와 비동기 Worker 구조를 갖추고 있어 대량 문서 처리, 외부 시스템 연동, 모델 교체, 온프레미스 배포 등으로 확장하기 용이하다. 따라서 본 프로젝트는 한국형 문서 처리 기술의 실증 기반이자, 문서 기반 AI 서비스 생태계로 확장 가능한 기반 시스템으로 활용될 수 있다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

3 자기평가

본 프로젝트는 HWP/HWPX, PDF, 이미지, Excel 등 다양한 형식의 한국형 업무 문서를 AI 기반으로 분석하고, 이를 RAG 검색 및 질의응답에 활용 가능한 구조화 데이터로 변환하는 Document Parser 개발을 목표로 수행되었다. 최종 결과물은 React 기반 프론트엔드, FastAPI 기반 백엔드 API 서버, RabbitMQ 기반 비동기 작업 큐, Redis 상태 캐시, SQLite 저장소, OpenAI/OpenRouter/ Qwen2.5-VL 모델 연동 구조, WebSocket 기반 실시간 진행률 기능으로 구성되었다.

최종 결과물은 단순한 파일 업로드 기능이나 OCR 기능에 그치지 않고, 문서 업로드부터 변환 작업 생성, 비동기 처리, 진행 상태 확인, 결과 미리보기, 다운로드, 에러 로그 확인, RAG 질의응답까지 하나의 흐름으로 사용할 수 있도록 구현되었다. 특히 국내 업무 환경에서 많이 사용되는 HWP/HWPX 와 PDF 문서를 대상으로 텍스트, 표, 이미지 정보를 추출하고, 이를 Markdown, HTML table, JSON 메타데이터 형태로 정리하여 AI 활용 가능성을 높이고자 하였다.

자기평가는 기능 완성도, 사용성, 확장성, 안정성, 실무 적용 가능성, 유지보수성이라는 기준을 중심으로 수행하였다.

1. 기능 완성도 측면

기능 완성도 측면에서 본 프로젝트는 핵심 목표였던 문서 업로드, 변환 작업 생성, 변환 결과 조회, 문서 다운로드, 대시보드, 에러 로그, RAG 질의응답, WebSocket 실시간 진행 상태 확인 기능을 구현하였다. 또한 RabbitMQ 기반 비동기 작업 처리 구조를 적용하여 문서 변환 작업을 API 요청과 분리하였고, Worker 를 통해 실제 변환 작업을 수행하도록 구성하였다.

다만 RAG 기능의 경우 문서 기반 질의응답의 기본 흐름은 구현되었으나, 답변에 대한 참조 문서 표시나 근거 추적 기능은 아직 보완이 필요하다. 또한 일부 고급 문서 구조 요소, 예를 들어 결재선, 직인, 서식 필드, 체크박스, 서명 영역 등에 대한 정밀 인식은 향후 고도화 과제로 남아 있다. 따라서 전체 기능은 실사용 가능한 수준의 기반을 확보했지만, 고급 문서 이해 기능은 추가 개선이 필요하다고 판단한다.

2. 사용성 측면

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

사용자는 브라우저에서 파일을 업로드하고, 모델과 병렬 처리 옵션을 선택하며, 변환 상태를 확인하고, 결과를 미리보기 및 다운로드할 수 있다. 또한 WebSocket 기반 실시간 상태 전달을 통해 장시간 변환 작업의 진행 상황을 즉시 확인할 수 있도록 하였다.

이러한 구성은 개발자뿐만 아니라 일반 실무자도 사용할 수 있는 형태에 가깝다. 다만 실제 운영 환경에서는 사용자 권한 관리, 문서 검색 UX, 대량 파일 처리 시 화면 성능, 실패 작업 재시도 UI 등의 개선이 추가로 필요하다. 현재 결과물은 기본적인 업무 흐름을 수행하기에는 충분하지만, 상용 수준의 사용자 경험을 위해서는 세부 UI 개선이 필요하다.

3. 확장성 측면

확장성 측면에서는 비교적 좋은 구조를 확보하였다. 백엔드는 `api/`, `core/`, `worker/`, `db/`, `storage/` 등 역할별로 분리되어 있으며, API 요청 처리, 문서 변환 로직, 비동기 작업 실행, 데이터 저장 로직이 분리되어 있다. 또한 모델 카탈로그와 실행 백엔드 구조를 통해 OpenAI, OpenRouter, Qwen 계열 모델을 선택적으로 사용할 수 있도록 하였다.

RabbitMQ 기반 작업 큐와 Worker 구조는 향후 Worker 수를 늘리거나, OpenAI worker 와 Qwen worker 를 분리하여 확장하는 데 유리하다. Docker Compose 기반 실행 환경도 제공되어 개발 및 배포 환경을 통일할 수 있다. 따라서 향후 새로운 문서 형식, 새로운 VLM 모델, 추가 Worker, 온프레미스 GPU 노드 확장에 대응할 수 있는 기반은 마련되었다고 판단한다.

4. 안정성 측면

안정성 측면에서는 비동기 작업 큐, 작업 상태 저장, 실패 작업 재시도, 취소 처리, 에러 로그 조회 기능을 구현하여 기본적인 안정성을 확보하였다. HWP 변환 과정에서는 변환 실패 가능성을 고려하여 여러 변환 방식을 활용하고, PDF 페이지 및 결과 품질을 점검하는 방식으로 안정성을 높이고자 하였다.

다만 문서 변환은 입력 파일의 품질과 형식에 크게 영향을 받는다. 손상된 HWP 파일, 비정상적으로 변환되는 PDF, 복잡한 표 구조, 대용량 이미지가 포함된 문서에서는 여전히 실패 가능성이 존재한다. 또한 실제 대규모 운영 환경에서의 장시간 부하 테스트, 수십만 장 단위의 문서 처리 검증, GPU 메모리 안정성 검증은 추가로 필요하다. 따라서 안정성은 개발·검증 환경 기준으로는 확보되었으나, 대규모 운영 안정성은 향후 테스트가 필요하다.

5. 실무 적용 가능성 측면

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

실무 적용 가능성 측면에서 본 프로젝트는 충분한 가능성을 보인다. 공공기관과 기업에서 많이 사용하는 HWP/HWPX, PDF, 이미지, Excel 문서를 처리 대상으로 삼았고, 변환 결과를 RAG 와 검색 시스템에서 활용할 수 있는 구조로 제공했기 때문이다. 특히 문서 업로드, 변환, 결과 확인, 다운로드, 질의응답까지 하나의 웹 서비스 안에서 수행할 수 있어 실제 업무 흐름과의 연결성이 높다.

또한 OpenAI API 기반 처리와 Qwen2.5-VL 로컬 GPU 기반 처리 구조를 함께 고려하여, 일반 클라우드 환경과 보안이 중요한 온프레미스 환경 모두에 대응할 수 있는 방향성을 확보하였다. 이 점은 공공기관, 금융기관, 연구기관 등 외부 API 사용이 제한되는 환경에서 중요한 장점이 될 수 있다.

다만 실제 도입을 위해서는 기관별 문서 양식에 대한 추가 학습 또는 룰 보정, 권한 관리, 감사 로그, 개인정보 마스킹, 문서 보존 정책, 배포 보안 설정 등이 필요하다. 따라서 현재 결과물은 실무 적용을 위한 프로토타입과 기반 시스템으로는 충분하며, 운영 시스템으로 확장하기 위한 개선 여지도 명확하다고 평가한다.

6. 유지보수성 측면

유지보수성 측면에서는 프론트엔드와 백엔드가 분리되어 있고, 프론트엔드는 FSD 구조를 기반으로 `app`, `pages`, `widgets`, `features`, `entities`, `shared` 계층으로 구성되어 있다. 백엔드는 API, core 로직, worker, storage, db 가 분리되어 있어 기능별 수정 범위를 파악하기 쉽다. 또한 ESLint, Prettier, Husky, Vitest, Playwright, Storybook 등 품질 관리 도구를 도입하여 코드 품질을 유지할 수 있는 기반을 마련하였다.

다만 프로젝트가 빠르게 확장되면서 일부 레거시 코드와 사용하지 않는 모델 관련 코드가 남아 있으며, 문서와 실제 구현 사이에 일부 차이가 존재한다. 향후 유지보수성을 높이기 위해서는 사용하지 않는 코드 정리, 기술문서 최신화, API 스펙 정리, 테스트 커버리지 확대가 필요하다.

종합 평가

종합적으로 볼 때, 본 프로젝트는 한국형 업무 문서를 AI 가 활용 가능한 구조화 데이터로 변환하는 핵심 목표를 달성하였다. 문서 업로드, AI 기반 분석, 비동기 변환 처리, 실시간 진행률 확인, 결과 저장 및 조회, RAG 질의응답까지 전체 서비스 흐름을 구현했으며,

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

OpenAI 기반 클라우드 처리와 Qwen 기반 온프레미스 처리 가능성을 함께 고려한 점에서 확장성과 실무 적용 가능성이 높다.

물론 모든 기능이 상용 서비스 수준으로 완성된 것은 아니다. RAG 답변의 근거 문서 표시, 고급 문서 구조 인식, 대규모 성능 검증, 보안 운영 기능, 일부 레거시 코드 정리는 향후 개선 과제로 남아 있다. 그러나 최종 결과물은 단순 시연용 기능을 넘어 실제 문서 처리 업무에 적용 가능한 기반 구조를 갖추고 있으며, 후속 고도화를 통해 공공기관과 기업의 문서 기반 AI 활용 시스템으로 발전할 수 있다고 판단한다.

따라서 본 프로젝트의 결과물은 "사용 가능한 프로토타입이자 실무 적용을 위한 기반 시스템"으로 평가할 수 있다. 핵심 기능은 구현되었고, 구조적으로 확장 가능하며, 명확한 개선 방향도 확보되어 있다. 팀은 본 프로젝트를 통해 문서 처리, AI 모델 연동, 비동기 작업 처리, 웹 서비스 구현, 배포 환경 구성까지 전반적인 시스템 개발 역량을 확보할 수 있었다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

4 참고 문헌

번호	종류	제목	출처	발행년도	저자	기타
1	기사	AI 예산 10조 시대 열렸지만…"돈만 쓴다고 G3 되나"	https://industryjournal.co.kr/news/244757			
2	기사	민간 최신 AI, 행정망에서 보안 걱정 없이 활용해 'AI 행정시대' 활짝 연다	https://www.mois.go.kr/frt/bbs/type010/commonSelectBoardArticle.do?bbsId=BBSMSTR_000000000008&nttId=121943			
3	기술 문서	FastAPI - REST API 프레임워크	https://fastapi.tiangolo.com/			
4	기술 문서	React 19 Documentation	https://react.dev/			
5	기술 문서	TanStack Router (파일 기반 라우팅)	https://tanstack.com/router			
6	기술 문서	TanStack Query (서버 상태 관리)	https://tanstack.com/query			
7	기술 문서	shadcn/ui (UI 컴포넌트 라이브러리)	https://ui.shadcn.com/			
8	기술 문서	Feature-Sliced Design (FSD 아키텍처)	https://feature-sliced.design/			

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

9	기술 문서	OpenAI GPT-5.2 API	https://platform.openai.com/docs			
10	논문	DocLayNet: A Large Human-Annotated Dataset for Document-Layout Analysis	https://arxiv.org/pdf/2206.01062	Pfitzmann et al.	2022	
11	논문	"Qwen2.5-VL technical report,"	https://arxiv.org/abs/2502.13923	S. Bai et al.,	2025	
12	온라인 자료	"flash-attention: Fast and memory-efficient exact attention," GitHub repository.	https://github.com/Dao-AILab/flash-attention	Dao-AILab	2026	

5 부록

5.1 사용자 매뉴얼

사용자 매뉴얼은 LLMong 서비스를 실제로 사용하는 일반 사용자 관점에서 작성한다. 사용자는 별도의 명령어 실행이나 서버 설정 없이 웹 브라우저를 통해 서비스에 접속하고, 문서를 업로드한 뒤 변환 결과를 확인한다. 따라서 본 절에서는 설치나 배포 방법이 아니라 로그인, 문서 변환, 결과 확인, RAG 질의응답, 파일 관리, 에러 확인과 같은 화면 사용 절차를 중심으로 설명한다.

1. 서비스 접속 및 로그인

사용자는 웹 브라우저에서 LLMong 프론트엔드 주소에 접속한다. 로그인 화면에서 관리자에게 발급받은 계정 정보를 입력하여 서비스에 로그인한다. LLMong은 민감한 업무 문서를 다루는 시스템이므로 사용자가 임의로 회원가입하는 구조가 아니라, 관리자가 사용자 계정을 생성하고 권한을 부여하는 방식으로 운영된다.

최초 사용자는 발급받은 임시 비밀번호로 로그인한 뒤, 보안을 위해 비밀번호를 변경한다. 로그인 후에는 대시보드 화면으로 이동하며, 사용자는 문서 변환, 파일 관리, RAG 질의응답, 에러 로그 화면을 사용할 수 있다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

2. 대시보드 확인

대시보드는 시스템의 전체 문서 처리 현황을 요약해서 보여주는 화면이다. 사용자는 전체 작업 수, 완료된 작업 수, 진행 중인 작업 수, 실패한 작업 수를 확인할 수 있다. 또한 일별 처리량, 성공률, 최근 작업 내역, 시스템 상태 정보를 통해 현재 문서 변환 서비스가 정상적으로 동작하는지 파악할 수 있다.

실패 작업이 증가하거나 특정 문서가 처리되지 않는 경우, 사용자는 에러 로그 화면 또는 파일 상세 화면으로 이동하여 원인을 확인할 수 있다.

3. 문서 업로드 및 변환 작업 생성

문서 변환 화면에서 사용자는 변환할 문서를 업로드한다. 파일 선택, 폴더 선택, 드래그 앤 드롭 방식의 업로드를 지원하며, 여러 개의 문서를 한 번에 등록할 수 있다. 지원되는 문서 형식은 HWP, HWPX, PDF, 이미지, Excel 등이다.

업로드 후 사용자는 변환에 사용할 모델과 병렬 처리 옵션을 선택한다. 모델 선택은 시스템 설정에 따라 OpenAI 기반 모델, OpenRouter 기반 모델, Qwen 계열 모델 중에서 사용할 수 있다. 병렬 처리 수는 여러 문서를 동시에 처리할 때 사용되며, 서버 성능과 작업량을 고려하여 설정한다.

설정을 완료한 뒤 변환 작업을 생성하면, 시스템은 문서를 작업 단위로 등록하고 비동기 Worker를 통해 변환을 수행한다.

4. 변환 진행 상태 확인

문서 변환이 시작되면 사용자는 작업의 진행 상태를 화면에서 확인할 수 있다. LLMong은 WebSocket 기반 진행 이벤트를 통해 작업의 대기, 시작, 진행, 완료, 실패, 취소 상태를 실시간으로 전달한다.

사용자는 각 파일이 현재 처리 중인지, 완료되었는지, 실패했는지 확인할 수 있으며, 실패한 작업은 에러 로그 또는 상세 화면에서 원인을 확인할 수 있다. 장시간 처리되는 문서의 경우에도 진행 상태가 표시되므로 사용자는 작업이 정상적으로 진행 중인지 파악할 수 있다.

5. 변환 결과 확인

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

변환이 완료된 문서는 결과 화면에서 확인할 수 있다. 사용자는 원본 문서 미리보기와 변환 결과를 함께 비교할 수 있으며, 추출된 텍스트, 표, 이미지 설명, 메타데이터를 확인할 수 있다.

결과 화면은 Preview, HTML, JSON 등의 보기 방식을 제공한다. Preview에서는 사람이 읽기 쉬운 형태로 변환 결과를 확인하고, HTML에서는 표 구조를 HTML table 형태로 확인할 수 있으며, JSON에서는 문서 메타데이터와 구조화된 결과를 확인할 수 있다.

필요한 경우 사용자는 변환 결과 파일을 다운로드하여 외부 시스템이나 후속 작업에 활용할 수 있다.

6. RAG 질의응답 사용

RAG 화면에서는 변환된 문서를 기반으로 자연어 질의응답을 수행할 수 있다. 사용자는 문서 내용을 직접 열람하지 않아도 질문을 입력하여 필요한 정보를 검색할 수 있다.

시스템은 변환된 문서 내용을 검색 가능한 단위로 분할하고, 사용자의 질문과 관련된 문서 내용을 찾아 답변을 생성한다. 이를 통해 사용자는 긴 문서나 여러 문서에서 필요한 정보를 빠르게 확인할 수 있다.

7. 파일 관리

파일 관리 화면에서는 업로드된 문서와 변환 결과 파일을 확인할 수 있다. 사용자는 파일명, 처리 상태, 변환 모델, 생성 시각 등을 기준으로 문서 목록을 확인하고, 필요한 문서의 상세 화면으로 이동할 수 있다.

변환이 완료된 파일은 미리보기 또는 다운로드가 가능하며, 더 이상 필요하지 않은 파일은 삭제할 수 있다. 파일 삭제 시에는 원본 문서와 변환 결과가 함께 정리될 수 있으므로 삭제 대상이 맞는지 확인한 뒤 수행한다.

8. 에러 로그 확인

에러 로그 화면에서는 문서 변환 과정에서 발생한 실패 내역을 확인할 수 있다. 사용자는 실패한 파일명, 에러 유형, 발생 시각, 상세 메시지를 확인할 수 있으며, 에러 유형별 통계도 확인할 수 있다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

에러 상세 정보에는 변환 실패 원인, 관련 Job 정보, Job Item 정보, 원본 오류 메시지 등이 포함된다. 사용자는 이를 바탕으로 파일 손상, 지원하지 않는 형식, API 키 문제, 모델 응답 실패, 변환 도구 오류 등을 확인하고 필요한 조치를 취할 수 있다.

9. 사용자 이용 흐름 요약

사용자는 다음 순서로 LLMong을 사용할 수 있다.

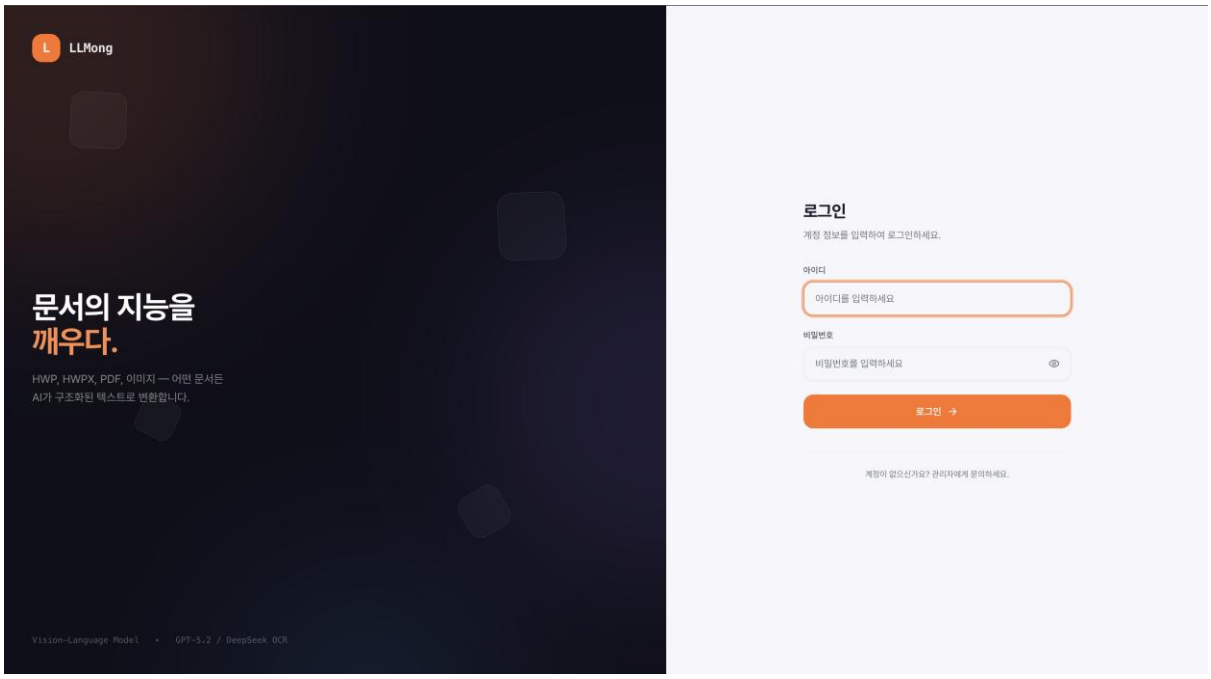
1. 웹 브라우저에서 서비스 접속
2. 발급받은 계정으로 로그인
3. 문서 변환 화면에서 파일 업로드
4. 변환 모델 및 병렬 처리 옵션 선택
5. 변환 작업 생성
6. 진행 상태 확인
7. 변환 결과 미리보기 및 다운로드
8. 필요한 경우 RAG 질의응답 수행
9. 파일 관리 또는 에러 로그 확인

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

[서비스 플로우]

1.로그인 화면

- LLMong은 민감한 문서를 다루는 시스템이므로 일반 사용자가 자유롭게 회원가입하는 구조가 아니라, 관리자가 사용자를 생성하고 권한을 부여하는 방식으로 설계하였다. 최초 실행 시 운영자는 환경 변수에 설정된 관리자 계정으로 bootstrap 절차를 수행하여 SUPERUSER 계정을 생성한다. 이후 관리자는 사용자 관리 화면에서 일반 사용자 또는 추가 관리자 계정을 생성할 수 있으며, 사용자는 발급받은 임시 비밀번호로 로그인한 후 비밀번호를 변경하여 서비스를 이용한다.



2. 대시보드 화면 설명

대시보드 화면은 LLMong 시스템의 전체 문서 처리 현황을 한눈에 확인하기 위한 화면이다. 운영자 또는 사용자는 대시보드를 통해 업로드된 문서 수, 변환 완료 문서 수, 처리 중인

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

작업 수, 실패한 작업 수 등을 확인할 수 있으며, 전체 시스템의 처리 상태를 빠르게 파악할 수 있다.

대시보드 상단에는 주요 작업 상태 요약 카드가 표시된다. 이 영역에서는 전체 작업 수, 완료된 작업 수, 진행 중인 작업 수, 실패한 작업 수와 같은 핵심 지표를 제공한다. 이를 통해 현재 시스템에 등록된 문서 변환 작업이 정상적으로 처리되고 있는지, 실패 작업이 증가하고 있는지 즉시 확인할 수 있다.

또한 대시보드에는 일별 처리량 트렌드 차트가 제공된다. 이 차트는 날짜별로 업로드되거나 처리된 문서 수를 시각화하여, 특정 기간 동안 문서 처리량이 어떻게 변화했는지 확인할 수 있도록 한다. 이를 통해 사용량 증가 추세나 특정 시점의 작업 집중 현상을 파악할 수 있다.

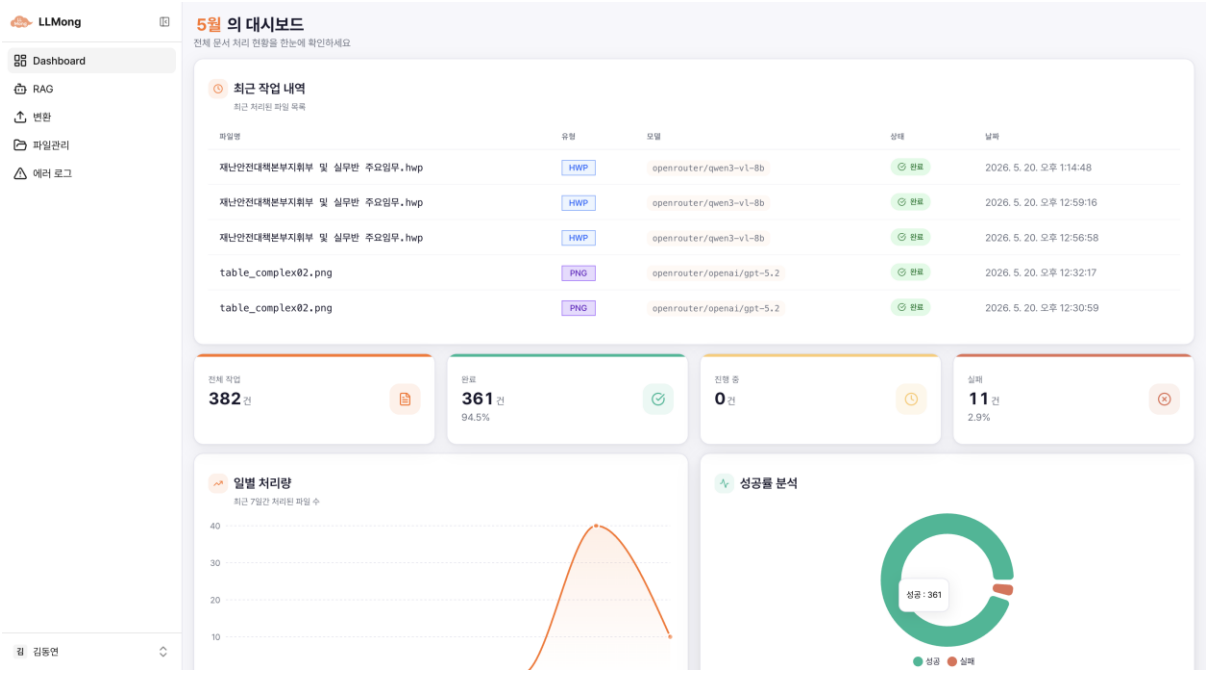
성공률 통계 영역에서는 전체 변환 작업 중 성공한 작업과 실패한 작업의 비율을 확인할 수 있다. 성공률은 문서 파싱 파이프라인의 안정성을 판단하는 기준으로 활용되며, 실패율이 높아지는 경우 에러 로그 화면에서 원인을 추가로 확인할 수 있다.

최근 작업 내역 영역에서는 최근 업로드되거나 변환된 문서 목록을 확인할 수 있다. 각 항목에는 파일명, 처리 상태, 업로드 시각 또는 갱신 시각 등이 표시되며, 사용자는 최근 작업이 완료되었는지, 실패했는지 빠르게 확인할 수 있다.

시스템 모니터링 영역에서는 CPU load, 저장소 사용량, 작업 큐 상태 등 시스템 운영에 필요한 정보를 확인할 수 있다. 이를 통해 문서 변환 작업이 서버 자원에 미치는 영향을 파악하고, 작업이 과도하게 쌓이거나 시스템 부하가 증가하는 상황을 점검할 수 있다.

결과적으로 대시보드 화면은 문서 변환 서비스의 운영 상태를 요약해서 보여주는 관리 화면이다. 사용자는 이 화면을 통해 전체 작업 현황, 처리량 변화, 성공률, 최근 작업, 시스템 상태를 확인하고, 문제가 발생한 경우 에러 로그나 문서 상세 화면으로 이동하여 원인을 분석할 수 있다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.



3.문서 변환 페이지 설명

문서 변환 페이지는 사용자가 문서를 업로드하고 AI 기반 변환 작업을 생성하는 핵심 화면이다. 사용자는 이 화면에서 HWP/HWPX, PDF, 이미지, Excel 등 지원되는 문서를 업로드하고, 변환 모델과 병렬 처리 옵션을 설정한 뒤 문서 파싱 작업을 실행할 수 있다.

화면의 업로드 영역에서는 파일 선택, 폴더 선택, 드래그 앤 드롭 방식의 업로드를 지원한다. 사용자는 단일 파일뿐만 아니라 여러 개의 파일을 한 번에 추가할 수 있으며, 폴더 단위 업로드를 통해 대량 문서를 변환 목록에 등록할 수 있다. 업로드된 파일은 목록 형태로 표시되며, 각 파일의 이름, 크기, 상태를 확인할 수 있다.

변환 설정 영역에서는 사용할 VLM 모델과 병렬 처리 수를 선택할 수 있다. 모델 선택 기능을 통해 OpenAI 기반 모델, OpenRouter 기반 모델, Qwen 계열 로컬 모델 등 시스템에서 제공하는 모델을 선택할 수 있으며, 병렬 처리 수를 조정하여 여러 문서를 동시에 처리할 수 있다. 이를 통해 사용자는 문서의 중요도, 처리 속도, 서버 자원 상황에 맞게 변환 방식을 선택할 수 있다.

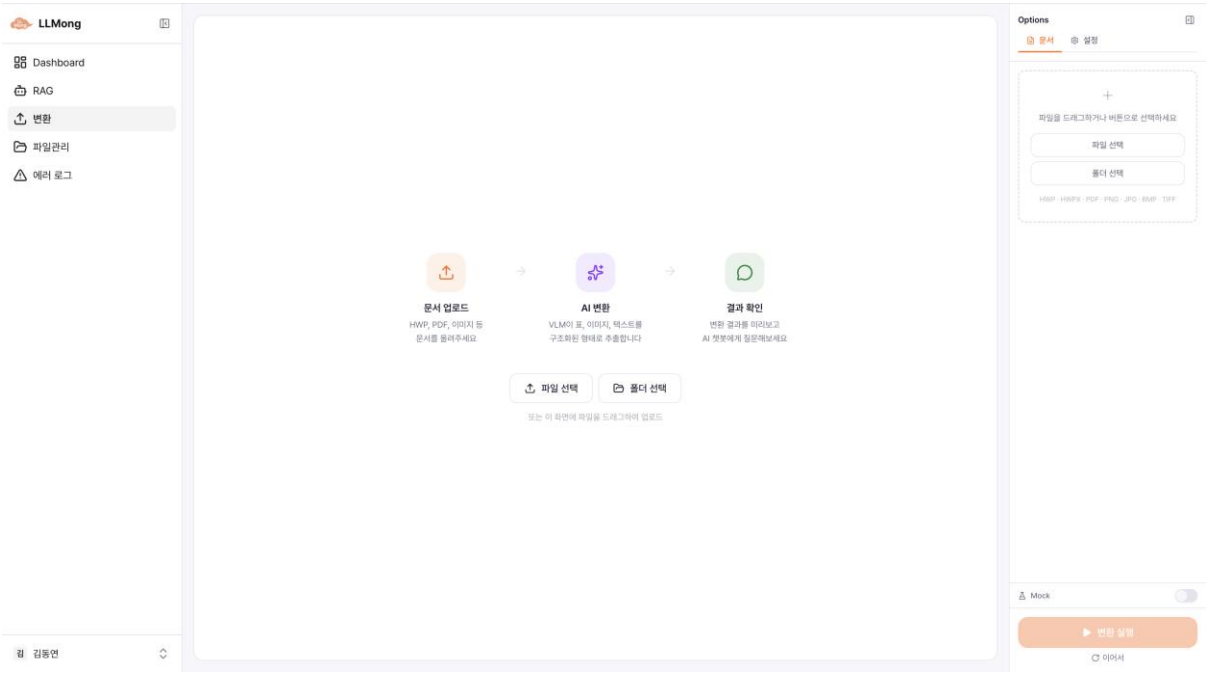
사용자가 변환을 실행하면 프론트엔드는 백엔드에 문서 변환 Job을 생성한다. 백엔드는 업로드된 파일을 저장하고, 각 파일을 Job Item으로 등록한 뒤 RabbitMQ 또는 메모리 큐에 작업을 적재한다. 이후 Worker가 작업을 가져와 실제 문서 변환을 수행한다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

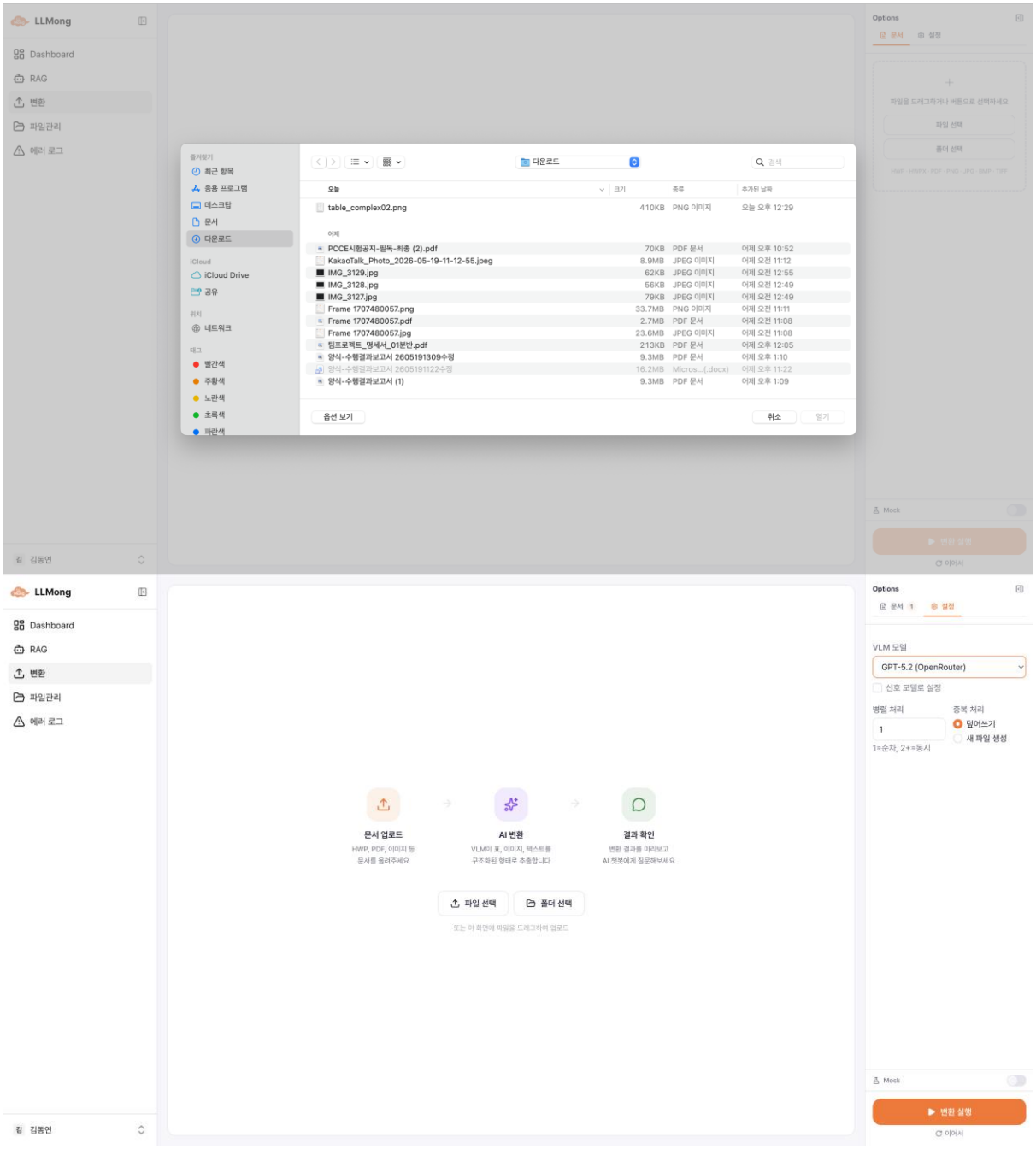
문서 변환 진행 중에는 각 파일의 처리 상태가 화면에 표시된다. 상태는 대기 중, 처리 중, 완료, 실패, 취소 등으로 구분되며, WebSocket 또는 진행률 API를 통해 실시간으로 갱신된다. 사용자는 변환이 정상적으로 진행되고 있는지, 어떤 파일이 실패했는지, 전체 작업이 어느 정도 완료되었는지 확인할 수 있다.

변환이 완료된 문서는 결과 영역에서 확인할 수 있다. 결과 미리보기에서는 추출된 텍스트, 표, 이미지 설명, Markdown 또는 HTML table 형태의 구조화 결과를 확인할 수 있으며, 필요한 경우 결과 파일을 다운로드할 수 있다. 실패한 문서의 경우 에러 메시지를 확인하고 재시도하거나 다른 모델로 다시 변환할 수 있다.

결과적으로 문서 변환 페이지는 LLMong의 주요 작업 흐름인 "문서 업로드 → 변환 설정 → 작업 생성 → 실시간 진행 상태 확인 → 결과 확인 및 다운로드"를 제공하는 중심 화면이다. 사용자는 별도의 CLI나 API 사용 없이 웹 브라우저에서 문서 파싱 작업을 수행할 수 있으며, 대량 문서 처리와 실시간 상태 확인을 통해 실무 환경에서도 효율적으로 문서 변환을 진행할 수 있다.



All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.



VLM 모델 선택 가능

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

실시간 진행률 확인 가능

4. 문서 변환 완료 후 결과 확인 화면 설명

문서 변환이 완료되면 사용자는 파일 관리 또는 변환 결과 화면에서 원본 문서와 파싱 결과를 함께 확인할 수 있다. 이 화면은 변환된 문서가 원본과 비교하여 얼마나 정확하게 구조화되었는지 검토하기 위한 결과 확인 화면이다.

화면 왼쪽에는 원본 문서 미리보기가 표시된다. 사용자는 업로드한 원본 이미지 또는 PDF 페이지를 확인할 수 있으며, 확대/축소, 페이지 이동, 화면 맞춤 기능을 통해 문서의 세부 내용을 검토할 수 있다. 이를 통해 변환 결과가 원본 문서의 표, 항목, 체크박스, 텍스트 구조를 제대로 반영했는지 비교할 수 있다.

화면 오른쪽에는 Document parsing 결과가 표시된다. 이 영역은 AI 기반 문서 파싱 파이프라인이 원본 문서를 분석하여 생성한 구조화 결과를 보여준다. 표 형태의 문서는 행과 열 구조가 유지된 상태로 표시되며, 문서 안의 제목, 구분 항목, 세부 내용, 체크박스 정보 등이 파싱 결과에 반

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

영된다.

상단에는 결과 보기 방식을 선택할 수 있는 탭이 제공된다. Preview 탭에서는 사용자가 읽기 쉬운 형태로 파싱 결과를 확인할 수 있으며, HTML 탭에서는 HTML table 기반의 구조화 결과를 확인할 수 있다. JSON 탭에서는 문서 메타데이터와 파싱 결과를 기계가 처리하기 쉬운 구조화 데이터 형태로 확인할 수 있다.

이 화면의 핵심 목적은 원본 문서와 변환 결과를 나란히 비교하는 것이다. 사용자는 왼쪽의 원본 문서를 보면서 오른쪽의 파싱 결과가 텍스트와 표 구조를 올바르게 추출했는지 확인할 수 있다. 특히 표, 병합 셀, 항목 구분, 체크박스와 같이 단순 OCR만으로는 구조가 깨지기 쉬운 요소가 어떻게 복원되었는지 검토할 수 있다.

또한 변환 상태가 완료된 문서는 상단에 변환 완료 상태로 표시되며, 사용자는 이후 결과를 다운로드하거나 RAG 검색 및 질의응답에 활용할 수 있다. 즉, 이 화면은 문서 변환 결과의 품질을 검토하고, AI가 생성한 구조화 데이터를 실제 업무에 활용하기 전에 확인하는 검수 화면의 역할을 한다.

결과적으로 문서 변환 완료 후 결과 확인 화면은 "원본 문서 확인 → 파싱 결과 비교 → Preview/HTML/JSON 결과 검토 → 후속 활용"으로 이어지는 검증 중심 화면이다. 이를 통해 사용자는 변환 결과의 정확성을 직접 확인하고, 구조화된 문서 데이터를 검색, 분석, RAG 질의응답 등에 활용할 수 있다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

5. RAG 화면 설명

RAG 화면은 변환된 문서를 기반으로 사용자가 자연어 질문을 입력하고, 문서 내용에 근거한 답변을 받을 수 있는 질의응답 화면이다. 사용자는 문서 내용을 직접 일일이 열람하지 않아도 질문을 통해 필요한 정보를 검색할 수 있으며, 시스템은 변환된 문서의 텍스트와 구조화 데이터를 활용하여 관련 내용을 찾고 답변을 생성한다.

화면은 기본적으로 문서 기반 채팅 인터페이스 형태로 구성된다. 사용자는 질문 입력창에 알고 싶은 내용을 자연어로 입력하고 전송할 수 있다. 예를 들어 특정 문서의 사업명, 날짜, 금액, 주요 항목, 조건, 신청 정보 등을 질문하면, 시스템은 변환된 문서 내용을 검색하여 관련 정보를 기반으로 답변을 생성한다.

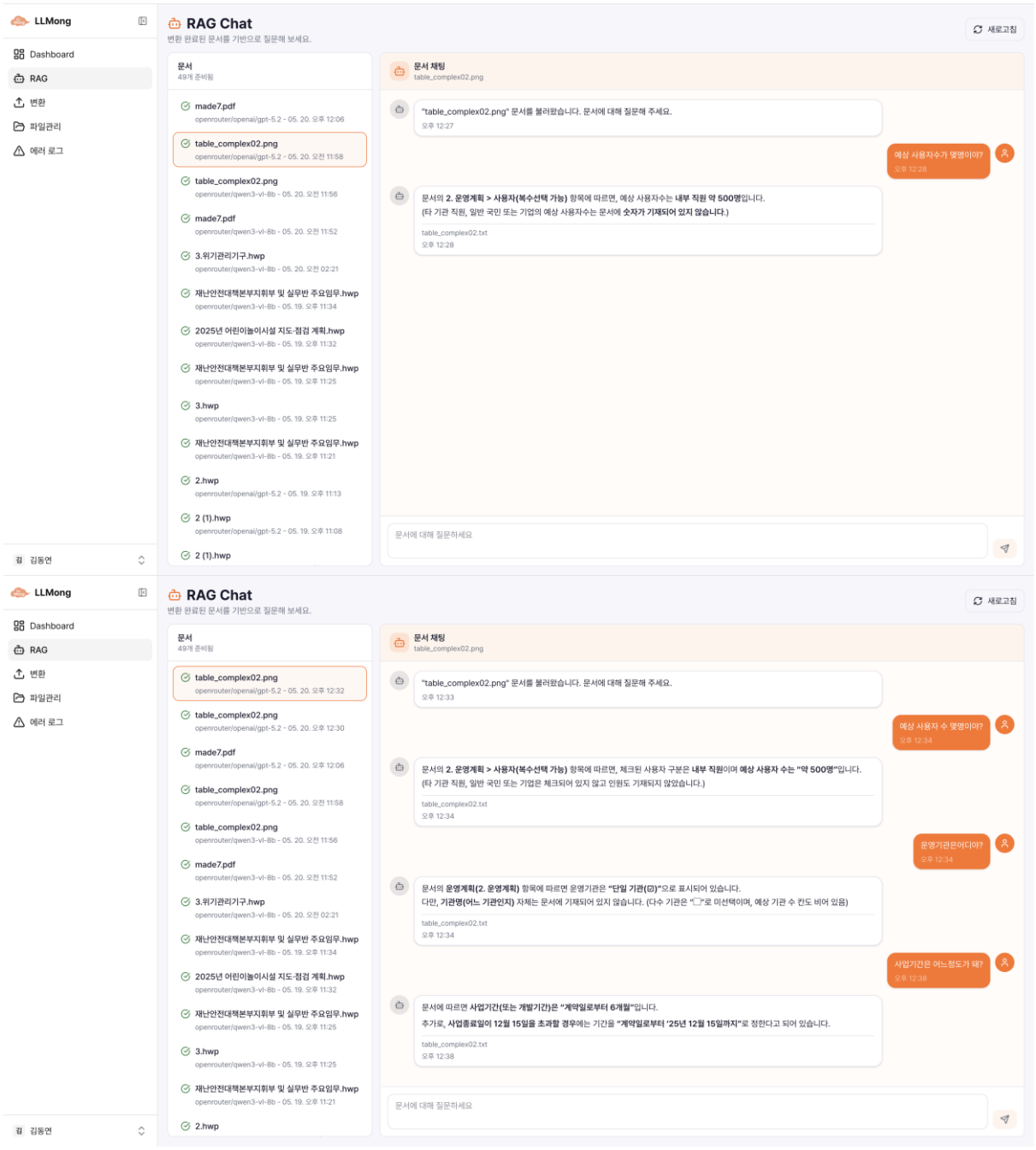
RAG 기능은 문서 변환 결과를 활용한다. 변환된 문서는 일반 텍스트, Markdown, HTML table, 메타데이터 형태로 저장되며, 질의응답 과정에서는 이 문서 내용을 검색 가능한 단위로 나누어 사용한다. 사용자의 질문이 입력되면 시스템은 질문과 관련성이 높은 문서 chunk를 찾고, 해당 내용을 기반으로 LLM이 답변을 생성한다.

이 화면의 목적은 단순 키워드 검색이 아니라 문서 기반 질의응답을 제공하는 것이다. 사용자는 문서 안의 정확한 표현을 몰라도 자연어로 질문할 수 있으며, 시스템은 문서 내용에 기반하여 답변을 제공한다. 이를 통해 긴 문서를 직접 읽고 필요한 정보를 찾는 시간을 줄일 수 있다.

RAG 화면은 내부 문서 검색, 업무 매뉴얼 질의응답, 행정문서 확인, 계약서 내용 질의, 보고서 요약 등 다양한 업무에 활용될 수 있다. 특히 여러 문서를 대상으로 정보를 찾거나, 표와 문단에 흩어져 있는 내용을 확인해야 하는 경우 문서 탐색 시간을 줄이는 데 효과적이다.

결과적으로 RAG 화면은 LLMong이 변환한 구조화 문서를 실제 AI 활용 단계로 연결하는 화면이다. 문서 파싱 결과를 단순 저장하는 데 그치지 않고, 사용자가 질문을 통해 문서 내용을 검색하고 이해할 수 있도록 지원함으로써 문서 기반 지식 활용성을 높인다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.



6. 파일 관리 페이지 설명

파일 관리 페이지는 사용자가 업로드한 문서와 변환 완료된 문서를 조회하고 관리하는 화면이다. 이 화면에서는 시스템에 등록된 문서 목록을 확인하고, 각 문서의 처리 상태, 파일명, 업로드 시각, 모델 정보, 변환 결과 등을 한눈에 파악할 수 있다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

문서 목록 영역에는 업로드된 파일들이 테이블 형태로 표시된다. 각 행에는 문서명, 파일 형식, 변환 상태, 사용 모델, 업로드 또는 수정 시각 등의 정보가 제공된다. 이를 통해 사용자는 어떤 문서가 변환 완료되었는지, 어떤 문서가 처리 중인지, 어떤 문서에서 오류가 발생했는지 쉽게 확인할 수 있다.

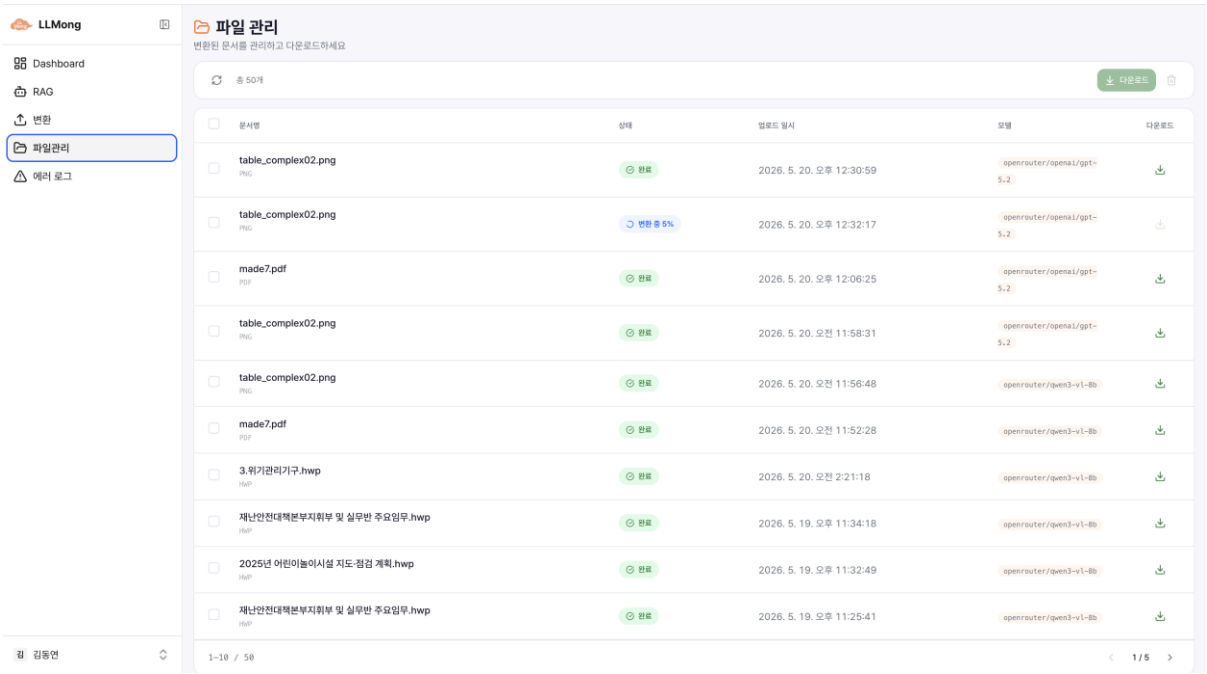
파일 관리 페이지에서는 변환 완료된 문서의 상세 결과로 이동할 수 있다. 사용자가 특정 문서를 선택하면 문서 상세 또는 결과 확인 화면에서 원본 문서와 파싱 결과를 비교하여 확인할 수 있다. 이때 Preview, HTML, JSON 등의 보기 방식을 통해 변환 결과를 다양한 형태로 검토할 수 있다.

또한 사용자는 변환 결과 파일을 다운로드할 수 있다. 다운로드 기능을 통해 TXT, 메타데이터 또는 원본 문서와 관련된 결과물을 로컬 환경에 저장할 수 있으며, 후속 보고서 작성, 데이터 검토, 외부 시스템 연동 등에 활용할 수 있다.

불필요한 문서는 삭제 기능을 통해 정리할 수 있다. 삭제 시 해당 문서의 파일 정보와 변환 결과, 관련 메타데이터가 함께 제거될 수 있으므로 사용자는 삭제 전 대상 문서를 확인해야 한다. 이 기능은 저장 공간 관리와 문서 목록 정리에 활용된다.

파일 관리 페이지는 단순한 파일 목록 화면이 아니라, 문서 변환 결과를 관리하고 후속 작업으로 연결하는 중심 화면이다. 사용자는 이 화면을 통해 문서 처리 상태를 확인하고, 결과를 열람하거나 다운로드하며, 필요 없는 문서를 삭제할 수 있다. 따라서 파일 관리 페이지는 LLMong의 문서 처리 흐름에서 변환 이후 결과 검토와 관리 역할을 수행한다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.



7. 에러 로그 페이지 설명

에러 로그 페이지는 문서 변환 과정에서 발생한 실패 내역과 오류 정보를 확인하기 위한 관리 화면이다. 사용자는 이 화면을 통해 어떤 문서에서 오류가 발생했는지, 오류 유형은 무엇인지, 언제 발생했는지, 어떤 작업과 연결되어 있는지를 확인할 수 있다.

에러 목록 영역에는 변환 실패 또는 경고 상태의 작업들이 테이블 형태로 표시된다. 각 항목에는 에러 발생 시각, 에러 유형, 심각도, 파일명, 작업 ID, 문서 ID, 처리 상태, 재시도 횟수 등의 정보가 제공된다. 이를 통해 사용자는 실패한 문서와 원인을 빠르게 파악할 수 있다.

에러 로그 페이지는 필터링 기능을 제공한다. 사용자는 에러 유형, 심각도, 파일명, 메시지 등을 기준으로 오류 목록을 검색하거나 필터링할 수 있다. 이를 통해 많은 작업 중 특정 유형의 오류만 확인하거나, 특정 문서에서 발생한 문제를 빠르게 찾을 수 있다.

에러 상세 정보 기능을 통해 개별 오류의 원인을 더 자세히 확인할 수 있다. 특정 에러 항목을 선택하면 에러 메시지, 원본 오류 payload, 관련 파일 정보, Job 정보, Job Item 정보, 권장 조치 사항 등을 확인할 수 있다. 이 정보는 파일 손상 여부 확인, 지원하지 않는 파일 형식 판단, 모델 변경 후 재처리, 작업 재시도 등의 조치를 결정하는 데 활용된다.

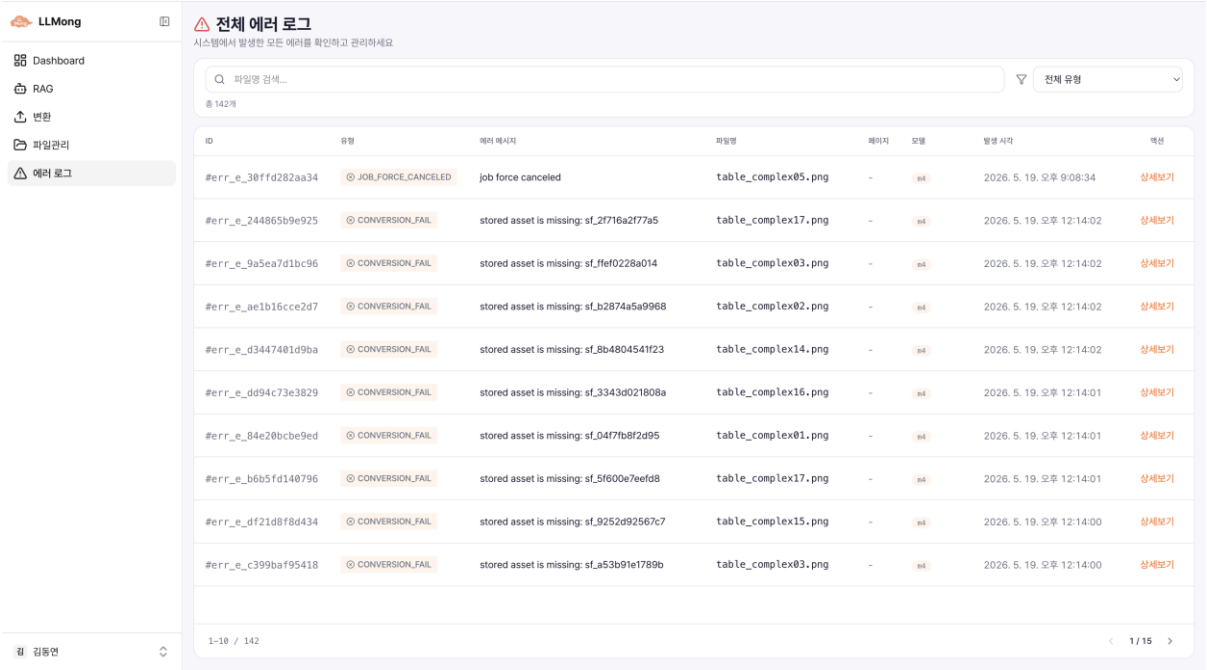
또한 에러 유형별 통계를 통해 반복적으로 발생하는 문제를 파악할 수 있다. 예를 들어 파일 변환 실패, API 호출 실패, 큐 처리 실패, 모델 응답 실패와 같은 오류가 많이 발생하는 경우, 운영

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

자는 해당 원인을 중심으로 시스템 설정이나 문서 처리 방식을 점검할 수 있다.

에러 로그 페이지는 단순히 실패 메시지를 보여주는 화면이 아니라, 문서 변환 시스템의 안정성을 관리하기 위한 운영 도구이다. 운영자와 사용자는 이 화면을 통해 실패 원인을 확인하고, 재시도 여부를 판단하며, 반복 오류를 분석하여 시스템 개선에 활용할 수 있다.

결과적으로 에러 로그 페이지는 LLMong의 문서 변환 과정에서 발생하는 문제를 추적하고 해결하기 위한 화면이다. 이를 통해 사용자는 실패한 문서를 방치하지 않고 원인을 파악할 수 있으며, 운영자는 전체 시스템의 오류 패턴을 분석하여 문서 처리 품질과 안정성을 높일 수 있다.



5.2 운영자 매뉴얼

운영자 매뉴얼은 LLMong 서비스를 관리하는 운영자 또는 관리자 관점에서 작성한다. 운영자는 사용자 계정 관리, 시스템 상태 점검, 에러 로그 확인, Worker 상태 확인, API 키 및 환경변수 관리, 장애 대응을 수행한다. 일반 사용자가 웹 화면에서 문서를 변환하는 것과 달리, 운영자는 서비스가 안정적으로 동작하도록 운영 환경을 관리하는 역할을 담당한다.

1. 관리자 계정 및 사용자 관리

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

LLMong 은 일반 사용자가 직접 회원가입하는 방식이 아니라, 관리자가 사용자를 생성하고 권한을 부여하는 방식으로 운영된다. 최초 실행 시 운영자는 환경변수에 설정된 관리자 계정 정보를 사용하여 SUPERUSER 계정을 생성한다.

관리자는 사용자 관리 화면에서 일반 사용자 또는 추가 관리자 계정을 생성할 수 있다. 새로 생성된 사용자는 임시 비밀번호로 로그인한 뒤 비밀번호를 변경한다. 운영자는 사용자의 역할과 권한을 관리하여 민감한 문서에 대한 접근을 제한할 수 있다.

2. 시스템 상태 확인

운영자는 대시보드와 모니터링 화면을 통해 전체 작업 현황을 확인한다. 완료, 진행 중, 대기, 실패 상태의 작업 수를 확인하고, 일별 처리량과 성공률을 통해 시스템이 정상적으로 운영되고 있는지 판단한다.

시스템 리소스 정보에서는 CPU load, 저장소 사용량, 작업 큐 상태 등을 확인할 수 있다. 작업이 과도하게 쌓이거나 실패율이 증가하는 경우 Worker 상태, 큐 상태, API 키 설정, 모델 응답 상태를 함께 점검한다.

3. Worker 및 작업 큐 확인

문서 변환 작업은 API 서버가 직접 처리하지 않고 Worker 가 비동기적으로 수행한다. 따라서 운영자는 Worker 가 정상적으로 실행 중인지 확인해야 한다.

작업이 대기 상태에 오래 머물러 있거나 변환이 진행되지 않는 경우 다음 항목을 점검한다.

- Worker 컨테이너 실행 여부
- RabbitMQ 큐에 작업이 적체되어 있는지 여부
- Worker 로그에 오류가 발생했는지 여부
- Redis 또는 SQLite 상태 저장소 접근 가능 여부
- OpenAI, OpenRouter, Qwen 모델 호출 실패 여부

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

RabbitMQ 관리 화면을 통해 큐에 쌓인 메시지 수와 소비 상태를 확인할 수 있으며, Worker 로그를 통해 실제 변환 중 발생한 오류를 확인할 수 있다.

4. API 키 및 환경변수 관리

운영자는 backend/.env 파일 또는 배포 환경의 환경변수를 관리한다. 주요 환경변수에는 관리자 계정 정보, 보안 키, OpenAI API 키, OpenRouter API 키, RabbitMQ 계정 정보, Redis 설정, Qwen 모델 경로 등이 포함된다.

운영 환경에서는 기본값을 사용하지 않고 충분히 긴 랜덤 문자열을 사용해야 한다. 특히 ADMIN_PW, ADMIN_UI_SECRET_KEY, APP_SECRET_KEY, RABBITMQ_PASSWORD 는 기본값을 그대로 사용하면 안 된다. API 키와 비밀번호가 Git 저장소에 커밋되지 않도록 주의한다.

5. 에러 로그 관리

운영자는 에러 로그 화면에서 변환 실패 내역을 확인한다. 에러 로그는 실패한 문서, 에러 유형, 발생 시각, 상세 메시지, 관련 Job 및 Job Item 정보를 제공한다.

에러 유형별 통계를 통해 반복적으로 발생하는 문제를 파악할 수 있다. 예를 들어 특정 파일 형식에서 실패가 반복되면 변환 파이프라인을 점검하고, API 호출 실패가 반복되면 API 키 또는 모델 제공자의 상태를 확인한다. Worker 처리 실패가 반복되면 큐 상태와 Worker 로그를 확인한다.

6. 장애 대응 절차

문서 변환이 정상적으로 수행되지 않을 경우 운영자는 다음 순서로 점검한다.

1. Backend health API 응답 확인
2. Frontend 접속 가능 여부 확인
3. RabbitMQ 실행 상태 확인
4. Redis 실행 상태 확인

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

5. Worker 실행 상태 및 로그 확인

6. API 키 및 모델 설정 확인

7. 실패 Job 및 에러 로그 확인

8. 필요한 경우 Worker 또는 Backend 재시작

9. 큐 적체 여부 확인

10. 동일 오류가 반복되면 입력 파일 또는 변환 파이프라인 점검

7. 보안 운영 주의사항

운영자는 민감한 문서와 API 키가 외부에 노출되지 않도록 관리해야 한다. 운영 배포 시 HTTPS 적용, 방화벽 설정, Redis/RabbitMQ 외부 노출 제한, 관리자 계정 최소화, 환경변수 보안 관리가 필요하다.

문서 파일에는 개인정보, 계약 정보, 행정 정보 등이 포함될 수 있으므로 접근 권한을 제한하고, 필요하지 않은 파일은 정기적으로 정리한다. 온프레미스 환경에서는 외부 API 호출 여부와 로컬 모델 사용 여부를 기관 보안 정책에 맞게 설정한다.

8. 운영자 점검 항목 요약

운영자는 다음 항목을 정기적으로 확인한다.

- Backend API 정상 동작 여부
- Frontend 접속 가능 여부
- Worker 실행 여부
- RabbitMQ 큐 적체 여부
- Redis 상태 확인
- 문서 변환 성공률
- 실패 작업 및 에러 로그
- API 키 유효성

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

- 저장소 용량
- 관리자 계정 및 사용자 권한
- 보안 설정 및 외부 포트 노출 여부

5.3 배포 가이드

배포 가이드는 LLMong 시스템을 개발 환경 또는 운영 환경에 설치하고 실행하는 절차를 설명한다. 배포 대상은 Frontend, Backend, Worker, Redis, RabbitMQ이며, 일반 실행 환경과 온프레미스 Qwen GPU 실행 환경을 구분하여 구성한다.

1. 배포 전 준비 사항

배포 전 다음 항목을 준비한다.

- Docker 및 Docker Compose
- Git
- OpenAI 또는 OpenRouter API 키
- 온프레미스 실행 시 NVIDIA GPU 및 CUDA 환경
- Qwen2.5-VL 모델 파일
- backend/.env 환경변수 파일
- 사용할 포트 정보

2. Repository Clone

프로젝트 저장소를 clone한 뒤 프로젝트 루트로 이동한다.

```
git clone https://github.com/kookmin-sw/capstone-2026-23.git cap_be
cd cap_be
```

실제 폴더명이 다를 경우 clone한 프로젝트 루트로 이동한다.

3. Backend 환경변수 설정

Docker Compose는 backend/.env 파일을 읽어 실행된다. 운영 환경에서는 관리자 계정, 보안 키, API 키, RabbitMQ 비밀번호 등을 반드시 변경해야 한다.

기본 개발용 예시는 다음과 같다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

ADMIN_ID=admin
ADMIN_PW=change-me-admin-password
ADMIN_UI_SECRET_KEY=change-me-admin-ui-secret-at-least-32-chars
APP_SECRET_KEY=change-me-app-secret-at-least-32-chars

AUTH_REQUIRED=true
OPENAI_API_KEY=your_openai_api_key
OPENAI_MODEL=gpt-5-mini

RAG_PROVIDER=openai
RAG_OPENAI_MODEL=gpt-4o-mini
RAG_OPENAI_EMBEDDING_MODEL=text-embedding-3-small

OpenRouter를 사용할 경우 다음 값을 추가한다.

OPENROUTER_API_KEY=your_openrouter_api_key
RAG_PROVIDER=openrouter
RAG_OPENROUTER_MODEL=openai/gpt-5-mini
RAG_OPENROUTER_EMBEDDING_MODEL=openai/text-embedding-3-small

운영 환경에서는 ADMIN_PW, ADMIN_UI_SECRET_KEY, APP_SECRET_KEY, RABBITMQ_PASSWORD를 기본값이 아닌 충분히 긴 값으로 교체한다.

4. Docker Compose 기반 일반 실행

일반 실행 환경에서는 Frontend, Backend, Worker, Redis, RabbitMQ를 Docker Compose로 함께 실행한다.

docker compose -f docker-compose.yml up -d --build

접속 정보는 다음과 같다.

Frontend: http://localhost:3000
Backend: http://localhost:8000
Swagger: http://localhost:8000/docs
Health: http://localhost:8000/v1/health
RabbitMQ: http://localhost:15672

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

실행 상태와 로그는 다음 명령으로 확인한다.

```
docker compose -f docker-compose.yml ps
docker compose -f docker-compose.yml logs -f backend
docker compose -f docker-compose.yml logs -f worker-openai
```

서비스 중지는 다음 명령으로 수행한다.

```
docker compose -f docker-compose.yml down
```

저장된 작업, 문서 상태, 볼륨까지 삭제해야 할 경우에만 다음 명령을 사용한다.

```
docker compose -f docker-compose.yml down -v
```

5. Backend 로컬 개발 실행

RabbitMQ 없이 API 동작을 빠르게 확인할 경우 backend/.env에 다음 개발용 값을 설정할 수 있다.

```
QUEUE_BACKEND=memory
STORE_BACKEND=sqlite
STATUS_CACHE_BACKEND=none
ENABLE_INLINE_EXEC_WORKER=true
ENABLE_INLINE_RECOVERY_WORKER=true
```

Linux/macOS 실행 예시는 다음과 같다.

```
cd backend
python3.12 -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip setuptools wheel
pip install -r requirements.txt
uvicorn api:app --host 0.0.0.0 --port 8000 --reload
```

Windows PowerShell 실행 예시는 다음과 같다.

```
cd backend
py -3.12 -m venv .venv
.₩.venv₩Scripts₩Activate.ps1
```


All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

```
python -m pip install --upgrade pip setuptools wheel
pip install -r requirements.txt
uvicorn api:app --host 0.0.0.0 --port 8000 --reload
```

API 문서는 다음 주소에서 확인한다.

http://localhost:8000/docs

6. Frontend 로컬 개발 실행

Frontend는 Vite 개발 서버를 사용한다. Backend가 localhost:8000에서 실행 중이어야 하며, Vite dev server는 /api 요청을 Backend로 프록시한다.

```
cd frontend
npm install
npm run dev
```

Frontend 개발 서버 주소는 다음과 같다.

Local dev: http://localhost:5173

다른 Backend 주소를 직접 지정해야 할 경우 frontend/.env.local에 다음 값을 설정한다.

```
VITE_API_BASE_URL=http://localhost:8000/api/v1
```

7. 온프레미스 Qwen GPU 실행

보안상 외부 API 사용이 어렵거나 로컬 GPU 기반 추론을 사용할 경우 온프레미스 Docker Compose 환경을 사용한다. Qwen2.5-VL-7B 모델을 사용할 경우 backend/.env 또는 shell 환경변수에 다음 값을 설정한다.

```
ENABLE_LOCAL_QWEN_MODEL=1
DEFAULT_AUTO_EXECUTION_BACKEND=qwen_gpu

QWEN_MODEL_HOST_DIR=./models
QWEN_VL_7B_MODEL_PATH=/models/Qwen2.5-VL-7B-Instruct
CUDA_VISIBLE_DEVICES=0
QWEN_INFER_WORKER_MAX_CONCURRENCY=1
QWEN_INFER_GPU_SLOTS=1
```


All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

GPU_MAX_CONCURRENT_INFERENCE=1

RABBITMQ_USER=llmong
RABBITMQ_PASSWORD=change-me-rabbitmq-password

모델 디렉터리는 기본적으로 ./models/Qwen2.5-VL-7B-Instruct를 컨테이너의 /models/Qwen2.5-VL-7B-Instruct로 마운트하여 사용한다.

온프레미스 환경 실행 명령은 다음과 같다.

docker compose -f docker-compose.onprem.yml up -d --build

포트가 이미 사용 중이면 .env 또는 shell 환경변수에서 FRONTEND_PORT, BACKEND_PORT, REDIS_PORT, RABBITMQ_PORT, RABBITMQ_MANAGEMENT_PORT를 변경한다.

8. 배포 후 검증

배포 완료 후 다음 항목을 확인한다.

- Frontend 접속 가능 여부
- Backend health API 정상 응답 여부
- Swagger 문서 접속 가능 여부
- 로그인 또는 bootstrap 정상 동작 여부
- 문서 업로드 가능 여부
- 변환 Job 생성 가능 여부
- Worker가 Job을 처리하는지 여부
- WebSocket 기반 진행 상태 표시 여부
- 변환 결과 미리보기 가능 여부
- 결과 파일 다운로드 가능 여부
- RAG 질의응답 동작 여부
- 에러 로그 화면 정상 동작 여부
- RabbitMQ 큐 적체 여부
- OpenAI 또는 Qwen 모델 호출 정상 동작 여부

9. 배포 요약

LLMong은 Docker Compose를 통해 Frontend, Backend, Worker, Redis, RabbitMQ를 함께 배포한다. 일반 환경에서는 OpenAI 또는 OpenRouter API 기반으로 문서 변환을 수행하

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

고, 온프레미스 환경에서는 Qwen2.5-VL 로컬 GPU Worker를 추가하여 내부망 또는 자체 GPU 서버에서도 문서 분석을 수행할 수 있다. 배포 후에는 Health API, Worker 로그, RabbitMQ 큐 상태, 문서 변환 기능, WebSocket 진행 상태, RAG 기능을 확인하여 정상 동작 여부를 검증한다.

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

5.4 테스트 케이스

대분류	소분류	기능	테스트 방법	기대 결과	테스트 결과
문서 변환	파일 업로드	단일 파일 선택 업로드	문서 업로드 화면에서 [파일 선택] 버튼을 클릭한 뒤 HWP, HWPX, PDF 또는 이미지 파일 1 개를 선택한다.	선택한 파일이 업로드 목록에 표시되고, 파일명과 파일 크기가 정상적으로 표시된다.	성공
문서 변환	파일 업로드	다중 파일 선택 업로드	[파일 선택] 버튼을 클릭한 뒤 여러 개의 문서 파일을 동시에 선택한다.	선택한 모든 파일이 업로드 목록에 추가되고, 각 파일이 개별 변환 대상으로 표시된다.	성공
문서 변환	파일 업로드	폴더 선택 업로드	[폴더 선택] 버튼을 클릭한 뒤 문서 파일이 포함된 폴더를 선택한다.	폴더 내부의 지원 파일들이 업로드 목록에 추가된다.	성공
문서 변환	파일 업로드	파일 드래그 앤 드롭	문서 파일을 업로드 영역으로 드래그 앤 드롭한다.	드래그한 파일이 업로드 목록에 추가된다.	성공
문서 변환	파일 업로드	폴더 드래그 앤 드롭	문서 파일이 포함된 폴더를 업로드	폴더 내부의 지원 파일들이 업로드 목록에 추가된다.	성공

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

			영역으로 드래그 앤 드롭한다.		
문서 변환	파일 형식 검증	지원 파일 형식 업로드	HWP, HWPX, PDF, PNG, JPG, BMP, TIFF 파일을 각각 업로드한다.	지원 형식의 파일이 정상적으로 업로드되고 변환 대상으로 등록된다.	성공
문서 변환	파일 형식 검증	미지원 파일 형식 업로드	지원하지 않는 확장자의 파일을 업로드한다.	파일이 변환 대상에서 제외되거나 처리 불가 상태로 표시된다.	성공
문서 변환	변환 설정	VLM 모델 선택	변환 설정에서 사용 가능한 VLM 모델을 선택한다.	선택한 모델 정보가 변환 요청에 반영된다.	성공
문서 변환	변환 설정	병렬 처리 수 설정	변환 설정에서 병렬 처리 수를 1 이상으로 변경한 뒤 변환을 실행한다.	설정한 병렬 처리 수가 변환 요청에 반영된다.	성공
문서 변환	변환 실행	단일 파일 변환	파일 1 개를 업로드한 뒤 변환 작업을 생성한다.	변환 Job 이 생성되고 작업 상태가 대기, 처리 중, 완료 순서로 변경된다.	성공
문서 변환	변환 실행	다중 파일 배치 변환	여러 파일을 업로드한 뒤 병렬 처리 옵션을	여러 파일에 대한 Job Item 이 생성되고 각	성공

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

			설정하고 변환 작업을 생성한다.	파일의 변환 상태가 개별적으로 관리된다.	
문서 변환	변환 결과	TXT 결과 생성	변환이 완료된 문서의 결과를 확인한다.	추출된 텍스트, 표, 이미지 설명이 포함된 TXT 결과가 생성된다.	성공
문서 변환	변환 결과	JSON 메타데이터 생성	변환 완료 후 문서 상세 또는 결과 메타데이터를 확인한다.	파일명, 모델, 상태, 생성 시각 등 변환 메타데이터가 저장된다.	성공
문서 변환	실패 처리	변환 실패 시 재시도	오류가 발생할 수 있는 파일 또는 실패 상태의 문서에 대해 재처리를 실행한다.	최대 재시도 횟수 범위 내에서 재처리가 수행되고, 실패 정보가 기록된다.	성공
배치 처리	작업 상태	Job 상태 조회	변환 작업 생성 후 Job 상태 조회 API 또는 화면에서 상태를 확인한다.	전체 작업 수, 완료 수, 실패 수, 진행 상태가 정상적으로 표시된다.	성공
배치 처리	작업 상태	Job Item 상태 조회	생성된 Job 의 개별 item 목록을 조회한다.	각 파일별 상태, 시작 시각, 완료 시각, 재시도 횟수가 표시된다.	성공

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

배치 처리	작업 취소	대기 중인 작업 취소	변환 작업 생성 후 완료되기 전에 취소 버튼 또는 cancel API 를 호출한다.	취소 요청이 반영되고 대기 중인 작업이 CANCELLED 상태로 변경된다.	성공
배치 처리	진행률	실시간 진행 상태 확인	변환 작업 실행 중 화면의 진행률 또는 상태 이벤트를 확인한다.	작업 시작, 진행, 완료, 실패 상태가 화면에 반영된다.	성공
파일 관리	목록 조회	문서 목록 조회	파일 관리 화면에 접속하여 업로드/변환된 문서 목록을 확인한다.	저장된 문서 목록이 파일명, 상태, 업로드 시각과 함께 표시된다.	성공
파일 관리	미리보기	변환 결과 미리보기	변환 완료된 문서를 선택하여 상세 화면 또는 결과 패널을 확인한다.	추출된 텍스트, 표, 이미지 분석 결과가 미리보기로 표시된다.	성공
파일 관리	다운로드	변환 결과 다운로드	변환 완료 문서의 다운로드 버튼을 클릭한다.	변환 결과 파일이 브라우저를 통해 다운로드된다.	성공
파일 관리	삭제	선택 문서 삭제	문서 목록에서 삭제할 문서를 선택한 뒤 삭제 기능을 실행한다.	선택한 문서와 관련 메타데이터가 삭제되고 목록에서 사라진다.	성공

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

대시보드	작업 현황	전체 작업 현황 조회	대시보드 화면에 접속한다.	전체 작업 수, 완료, 진행 중, 실패 작업 수가 표시된다.	성공
대시보드	통계	성공률 통계 조회	대시보드에서 성공률 차트를 확인한다.	완료/실패 데이터를 기반으로 성공률 통계가 표시된다.	성공
대시보드	통계	일별 처리량 조회	대시보드에서 일별 처리량 차트를 확인한다.	업로드 또는 처리 일자 기준으로 문서 처리량이 차트로 표시된다.	성공
대시보드	최근 작업	최근 작업 내역 조회	대시보드의 최근 작업 영역을 확인한다.	최근 처리된 문서 목록과 상태가 표시된다.	성공
에러 로그	목록 조회	변환 실패 내역 조회	에러 로그 화면에 접속한다.	실패한 작업 또는 경고 항목이 목록으로 표시된다.	성공
에러 로그	필터링	에러 유형별 필터링	에러 로그 화면에서 에러 유형 또는 심각도 필터를 적용한다.	선택한 조건에 해당하는 에러만 목록에 표시된다.	성공
에러 로그	통계	에러 유형별 통계 조회	에러 로그 요약 영역 또는 API 를 확인한다.	에러 유형별 발생 건수가 표시된다.	성공

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

에러 로그	상세 조회	에러 상세 정보 확인	에러 목록에서 특정 에러 항목을 선택한다.	에러 메시지, 파일명, 작업 ID, 문서 ID, 원본 에러 정보가 표시된다.	성공
RAG	세션	RAG 세션 생성	RAG 채팅 화면에서 새 질의응답 세션을 생성한다.	새로운 RAG 세션 ID 가 생성되고 채팅 화면이 초기화된다.	성공
RAG	질의응답	문서 기반 질문 입력	변환된 문서를 대상으로 질문을 입력하고 전송한다.	문서 내용을 기반으로 답변이 생성되어 화면에 표시된다.	성공
RAG	참조 문서	답변의 참조 문서 표시	RAG 답변 생성 후 citations 또는 참조 문서 영역을 확인한다.	답변에 사용된 참조 문서가 표시되어야 한다.	실패
인증	로그인	관리자 로그인	관리자 계정으로 로그인한다.	로그인 성공 후 access token 이 발급되고 관리자 기능에 접근할 수 있다.	성공
인증	사용자 관리	사용자 생성	관리자 권한으로 신규 사용자를 생성한다.	신규 사용자 계정이 생성되고 로그인 가능 상태가 된다.	성공
인증	비밀번호	비밀번호 변경	현재 비밀번호와 새 비밀번호를 입력하여 변경한다.	비밀번호가 변경되고 이후 새 비밀번호로 로그인할 수 있다.	성공

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

시스템 관리	Health Check	Backend 상태 확인	/v1/health API 를 호출한다.	Backend 서버의 정상 응답이 반환된다.	성공
시스템 관리	Queue	RabbitMQ 작업 큐 확인	RabbitMQ 관리 화면 또는 worker 로그에서 큐 상태를 확인한다.	생성된 작업이 큐에 적재되고 worker 가 작업을 소비한다.	성공
시스템 관리	Worker	Worker 처리 확인	문서 변환 작업 생성 후 worker 로그를 확인한다.	Worker 가 Job Item 을 가져와 변환 작업을 수행한다.	성공
시스템 관리	모니터링	CPU 및 저장소 상태 조회	시스템 모니터링 API 또는 대시보드에서 자원 상태를 확인한다.	CPU load, 작업 상태, 저장소 사용량이 표시된다.	성공
시스템 관리	모니터링	메모리 사용량 조회	시스템 모니터링 화면에서 메모리 사용량을 확인한다.	실제 메모리 사용량이 표시되어야 한다.	실패
배포	Docker 실행	Docker Compose 전체 실행	docker compose -f docker-compose.yml up -d --build 명령을 실행한다.	Frontend, Backend, Worker, Redis, RabbitMQ 컨테이너가 정상 실행된다.	성공
배포	Compose 검증	Compose 설정 검증	docker compose -f docker-compose.yml config --quiet 명령을 실행한다.	Compose 설정 오류 없이 검증이 완료된다.	성공

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

품질 관리	Frontend Lint	코드 스타일 검사	npm run lint 명령을 실행한다.	ESLint 검사 결과 오류가 없어야 한다.	성공
품질 관리	Frontend Build	프론트엔드 빌드	npm run build 명령을 실행한다.	TypeScript 및 Vite 빌드가 정상 완료된다.	성공
품질 관리	Frontend Test	단위 테스트 실행	npm run test 명령을 실행한다.	작성된 단위 테스트가 통과한다.	성공

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.