



「2025 IA x AI 해커톤」

개발 완료 보고서

팀 명 : TEAM_BMR

프로젝트명 : 부메랑

※ 유의사항

1. 본 보고서의 내용은 최대 2 page이내로 작성 (본 표지 제외)
2. 보고서의 설명을 보충하기 위해 필요한 사진 또는 그래프 첨부 가능
3. 제출 서류는 일체 반환을 하지 않음
4. 제출 파일명 작성 요령
 - 파일명: [2025 IA x AI 해커톤]_팀명
5. 서체: 맑은고딕, 크기: 12p, 줄간격: 160%
6. 제출처: 깃허브에 업로드

「2025 IA x AI 해커톤」

프로젝트명	부메랑 - 대화 듣고 서류 보고, 위험 알려주는 AI 부동산 메이트
프로젝트 목표	<p>청년들에게 부동산 계약은 여전히 낯설고 어렵다. 현장에서 중개인과 나눈 대화 속 약속이 실제 계약서에 반영되었는지, 계약서를 다 읽어보지 않으면 알기 어렵다.</p> <p>‘부메랑’은 현장 대화를 실시간으로 인식(STT)하고, 이후 계약서 내용을 함께 분석하여 대화 속 약속이 제대로 반영됐는지 검증하는 AI 메이트이다. AI는 이를 기반으로 종합 리포트를 생성하고, 누락·불일치·위험 항목을 표시해 사용자가 쉽게 확인하도록 돕는다.</p> <p>‘부메랑’은 법률 자문이나 사후 분쟁 대응이 아닌, 계약 현장에서 바로 신뢰를 확인하게 하는 실시간 동반자를 목표로 한다.</p>
개발 환경	README.md 참조
구현 기능	<p>1. 매물 체크리스트 기능 : 사용자가 매물을 보며 체크리스트를 생성해 항목(곰팡이, 수압, 채광 등)을 평가하고, 결과를 저장한다.</p> <p>2. 계약 대화 모니터링 기능 : 계약 현장에서 녹음한 음성을 실시간 STT로 인식하고, 즉시 텍스트 변환해 보여준다.</p> <p>3. 주요 문서 업로드 및 분석 기능 : 계약서, 등기부등본, 중개대상물확인서 등 핵심 문서를 OCR로 텍스트화하고, 이를 저장해 활용할 수 있도록 한다.</p> <p>4. AI 종합 리포트 기능 : LLM으로 대화 분석 결과, 문서 분석 결과, 매물 체크리스트 내용을 통합해 누락·불일치 항목과 조심해야 할 내용을 정리한 리포트를 자동 생성한다.</p>
코드 주요 설명	<p>1) STT 동작과정 WS /ws/record 녹음 화면 진입 시 웹소켓을 열고 WebRTC 음성 스트림을 받아 백엔드에서 코덱 변환→STT API 호출→응답 파싱 후 STT 텍스트를 DB에 저장.</p> <p>2) OCR 동작과정 POST /api/ocr/upload 사용자가 계약서를 업로드하면 OCR 파이프라인이 문서를 파싱해 구조화 데이터와 원문을 DB에 적재하고</p>

	<p>완료 이벤트를 웹소켓으로 푸시.</p> <p>3) LLM 동작과정</p> <p><i>POST /api/report/llm</i> 사용자의 종합 리포트 요청을 받아 LLM이 STT와 OCR 결과를 통합 분석해 보고서를 생성·저장하고, 처리 완료 웹소켓으로 알린 뒤 화면에 노출.</p>
개발 내용	<p>1) OCR (문서 분석 파이프라인)</p> <ul style="list-style-type: none"> - 목표 수준: 다양한 형식의 계약서를 자동 인식하고, 조항 단위로 구조화하는 OCR+LLM 기반 문서 파서 구축. - 기술적 완성도: 주택임대차표준계약서·등기사항전부증명서·중개대상물 확인설명서 등 핵심 문서를 완전 지원하며, 평균 응답 속도 20초 이내, OCR 적용 전 대비 4배 향상. - 문제 및 해결: OCR 인식률이 낮은 문서는 문단 단위 confidence score를 기준으로 LLM 후처리를 추가했고, 잘못 인식된 항목은 키워드 패턴 매칭으로 보정해 신뢰도 90% 이상 확보. <p>2) STT (실시간 음성 인식 파이프라인)</p> <ul style="list-style-type: none"> - 목표 수준: 계약 현장에서 실시간 음성을 인식해 텍스트로 변환하고, 대화 기반 리스크 분석의 입력으로 제공하는 MVP 구현. - 기술적 완성도: 평균 응답 속도 3초 이내, 음성 구간별 정확한 텍스트 매핑 완료. 화자 분리는 미구현 상태임. - 문제 및 해결: 소음 환경에서 오류율이 높았으나 ffmpeg 전처리(노이즈·울림 제거)를 통해 오류율을 크게 감소시켰다. 화자 구분 부재는 LLM Q&A 기반 맥락 분석으로 대체함.
기타 (선택)	<p>1) 역할 분담과 팀 구조</p> <ul style="list-style-type: none"> - 이상민 : 팀장 / LLM 에이전트 구성 및 프롬프팅 담당 - 양주영 : 문서 OCR 변환 및 PDF 파싱 담당 - 최연덕 : STT 및 프론트엔드 담당 - 최민제 : 인프라 / 클라우드 및 백엔드 담당 - 김준호 : 기획, 디자인 및 백엔드 담당 <p>2) 팀의 협업 원칙</p> <ul style="list-style-type: none"> - 각자 역할별로 독립 개발 후, API 및 데이터 스펙을 기준으로 연함 - 사전 타임라인에 따르며, 팀 전체 진행상황을 지속적으로 공유함 - 제한된 해커톤 일정 안에서 완벽한 구현보다는 기능 간 연관성 확보를 최우선 목표로 함



붙임1

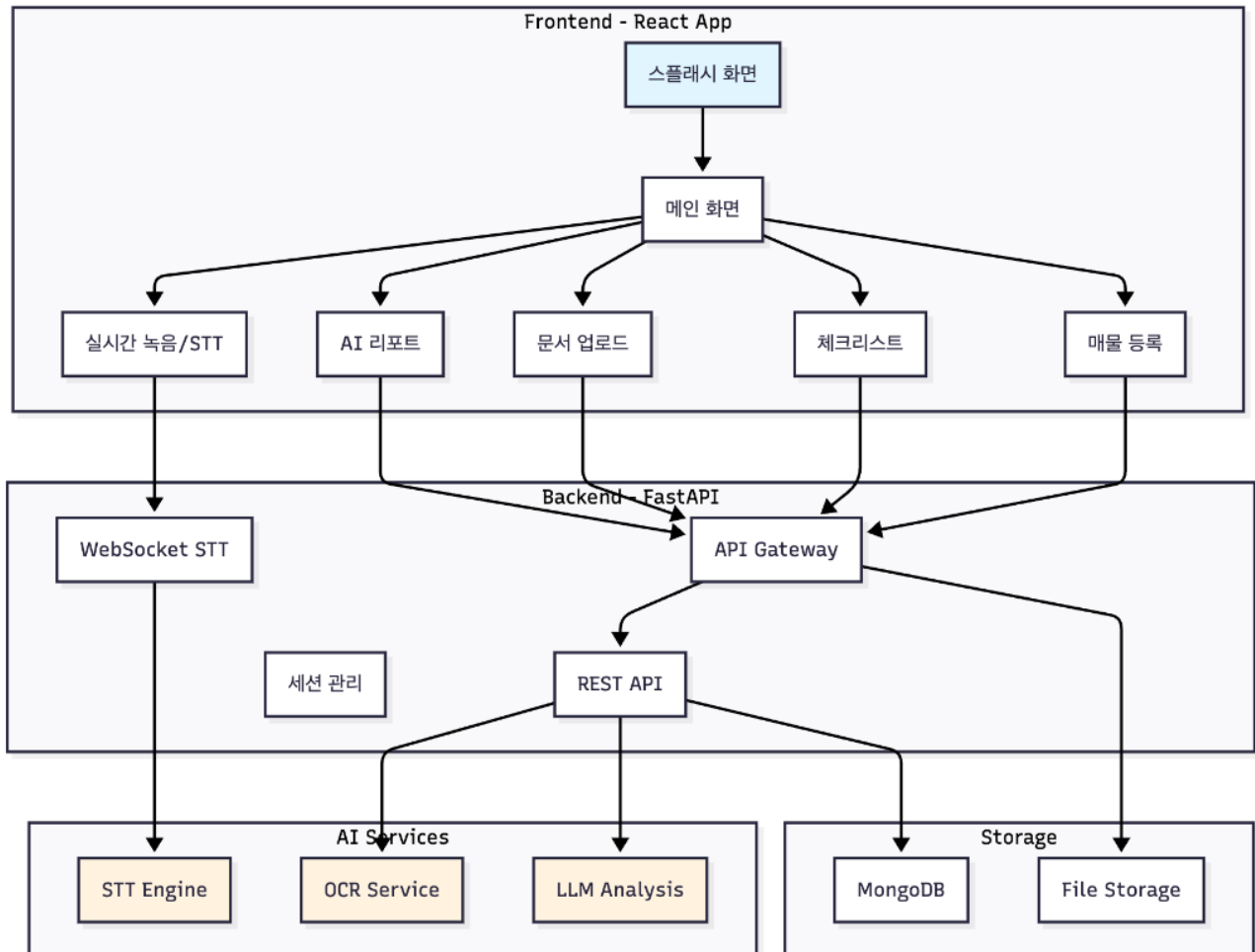
[팀 협업 타임라인]

	A	B	C	D	E
1	담당자	T0 (11:00 ~ 11:30)	T1 (11:30 ~ 13:59)	T2 (14:00 ~ 16:59)	T3 (17:00 ~ 19:59)
2	연덕	문서 및 초기 전략 수립 & 스키마 정의 & Endpoint 정의	프론트, 백엔드 STT		DB 데이터 가공
3	민초		인프라	백엔드	DB (MongoDB) / S3
4	상민		페르소나 4명	API 및 DB연결	프롬프팅 수정
5	주영		OCR 연결	파싱 1	파싱 2
6	준킴		문서 작성 전략 수립 및 문서 초안 작성	프론트 CSS 작업	프론트 CSS 작업 기존 문서 보완
7					
중간 점검		T4 (20:00 ~ 22:00)	T5 (22:00 ~ 01:00)	T6 (01:00 ~ 04:00)	
진행상황 공유 및 재개	백엔드 협업		STT와 나머지 백엔드 병합	프론트 UI 구현	
			퇴근		
	LLM + OCR 연결		비교 검증 깎기		
	자유 역할로 필요한 곳에 배정				
20:00 ~					
진행도					
STT 중 webRTC					
준킴님 백엔드 인수인계 완, S3(이미지), 몽고디비 연결까지 완, vercel이 레포권한때문에 포크해서 다른리포에서 프론트 빌드해야					
스키마 기반으로 어느정도 작동함. 화자구분 없는 컨텍스트로 꽤나게 데이터가 나오는 중. 페르소나 구현을 버전단위로 기록 중 프론트와 백엔드를 연결해서 결과를 보고 에이전트를 분리할 예정					
원문과 이미지 통째로 llm에 정해진 스키마 가지고 파싱을 진행 중이었는데 품질이슈로, 이미지를 ocr 결과로 뽑아내 llm에 파싱 함. 각 필드에 대한 성능을 덧붙여서 성능향상. ocr의 인식률이 문제가 있음. 단계적으로 고도화 됐고, 발전한 방향을 문서로 기록할 예정 그 뒤에 백엔드 연결하고 프론트에 붙을 예정					
백엔드 인수인계 교육완료. 합치고 나서 문서화 할 예정. 그래서 합치는게 중요하다. 전부 합치는 작업에 몰입해야 할 거 같다.					

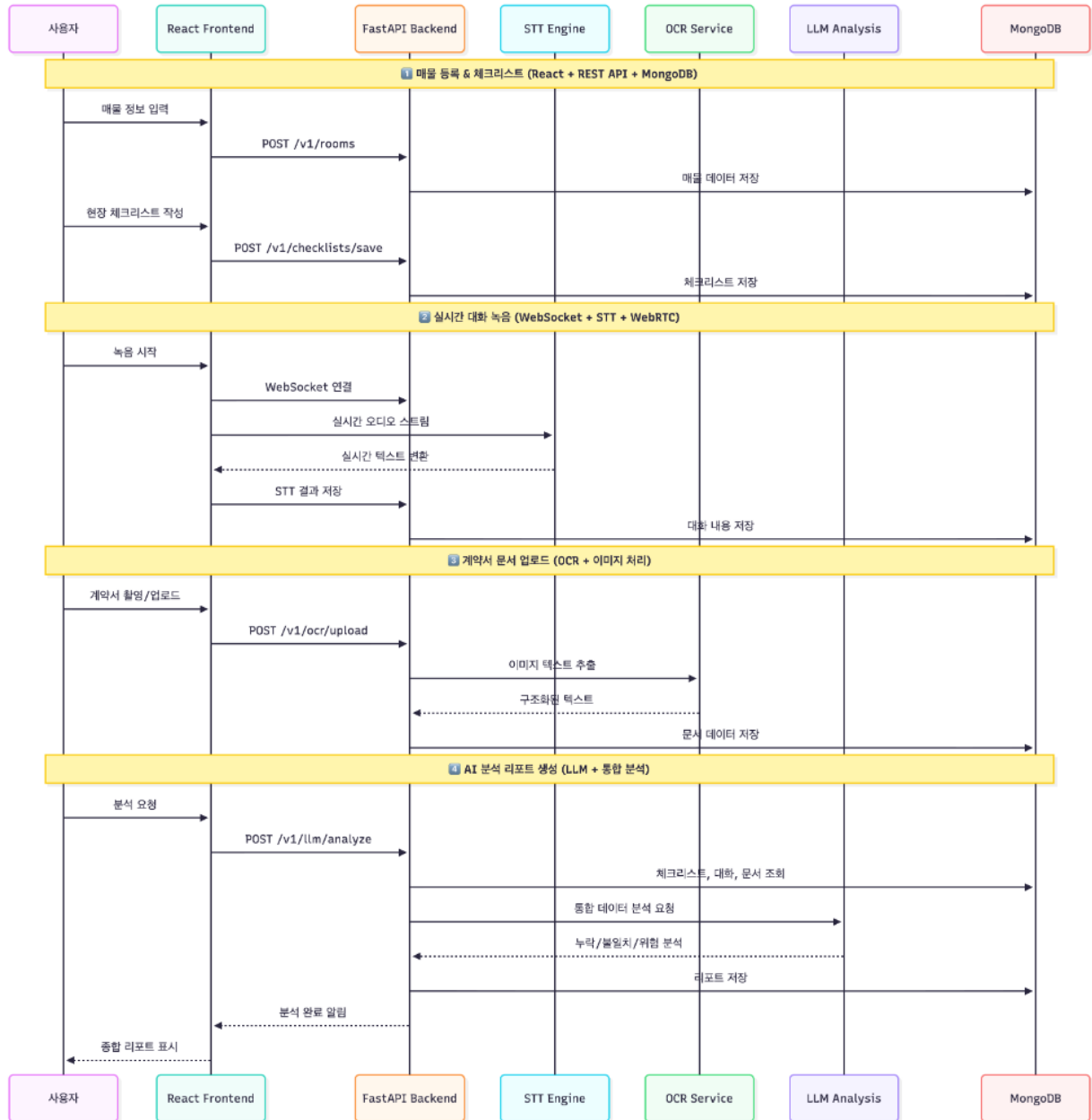
[부동산 현장 방문사진]



[시스템 다이어그램]



[시퀀스 다이어그램]



[Figma 협업]

