

**SK네트웍스 Family AI 과정 10기  
 모델배포 개발된 LLM 연동 웹 애플리케이션**



| **산출물 단계** | 모델배포 |
| --- | --- |
| **평가 산출물** | LLM 연동 웹 애플리케이션 |
| **제출 일자** | 2025.08.10 |
| **깃허브 경로** | [GitHub - SKNETWORKS-FAMILY-AICAMP/SKN12-FINAL-3TEAM](https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN12-FINAL-3TEAM) |
| **작성 팀원** | 이주영 |

### 1. 프로젝트 개요

* 프로젝트명: 회의 분석 기반 지능형 AI 프로젝트 관리 시스템, 딸깍 (TtalKkak)
* 작성일: 2025. 08. 08
* 작성자 : 이주영

### 2. 시스템 개요

* 목표: 이 애플리케이션은 AI 기반 업무 관리 플랫폼으로, 회의 분석부터 업무 생성, 팀 협업까지 모든 업무 프로세스를 통합 관리하며, 사용자가 효율적이고 체계적으로 업무를 수행할 수 있도록 합니다.
* 기능:
  + PostgreSQL 벡터 DB와 LLM 연동한 음성 분석 모델
  + 성격 유형별 효율적인 프롬프트 최적화
  + 통합 플랫폼 연동 및 웹 대시보드
  + 안전한 API 키 관리 및 예외 처리

### 3. 시스템 아키텍처

* 백엔드:  
  + 언어: TypeScript/JavaScript ([Node.js](http://node.js))
  + 프레임워크: Express.js 4.18.2
  + LLM 연동: RunPod AI 서버 (WhisperX, Triplet, BERT)
  + 데이터베이스: PostgreSQL + Prisma 5.1.1 + Redis 4.6.7
  + 벡터 데이터베이스: PostgreSQL + 외부 AI 서버
  + 기술 스택:
* JWT
* OAuth 2.0
* Winston
* Slack/Notion/Jira API
* 프론트엔드

### 프레임워크: React 18 + TypeScript

### 상호작용: 실시간 양방향 통신 및 상태 관리

### UI: TailwindCSS + Radix UI + Framer Motion

## 4. 기본 사용법

1. **랜딩페이지 설명**

* 상단부터 문제 인식 → 솔루션 → 성과 → 연동 서비스 → 후기 → FAQ 순서 확인
* 시작 버튼  
   • “Go to TtalKkak”과 로그인 버튼 : 클릭 후 OAuth 로그인 → Slack 워크스페이스 연결  
   • “Go to Dashboard” : – 로그인 후 실제 서비스 연동  
   • “데모 체험하기” – 로그인 없이 대시보드 기능 미리보기

1. **웹 대시보드 사용법**

1. 메인 대시보드 (MainContent)

* 헤더: "오늘의 프로젝트 현황" 제목과 랜딩페이지 이동 버튼 확인
* 빠른 액션
  + 음성 파일 업로드: MP3, WAV 업로드 -> 자동 회의록 생성 (회의분석메뉴에서 확인)
  + 문서 분석: PDF, DOC, DOCX, TXT, PPT, PPTX, XLS, XLSX 파일 업로드
* 업무 현황 (칸반 보드)
  + 해야할 일: 대기중인 업무
  + 진행중: 진행 중인 업무
  + 완료: 완료된 업무
  + 업무 카드: 제목, 날짜, 담당자 표시 ( 수정 버튼 선택 시 수정 가능 )
  + 드래그 앤 드롭: 업무 카드를 드래그해서 상태 변경
  + 지연 업무: 날짜가 지난 업무는 빨간색으로 표시
  + 새 업무 추가: 모달을 통한 수동 업무 생성

2. 회의 분석 (MeetingAnalysis)

* 회의 목록: 프로젝트별 회의 기록 확인
* 검색 필터: 회의명, 날짜, 참가자별 필터링 -> 원하는 회의 선택
* 회의록
  + 회의 기본 정보, 목적, 주요 내용, 결정사항 표시
  + [수정] 버튼 클릭 → 내용 편집
  + [다운로드] 버튼 클릭 → Word 문서로 저장
* 파생 업무: 회의에서 도출된 업무 목록 표
* 체크리스트: 회의 후 해야 할 일들 관리 (로컬 스토리지 저장)
* 다음 회의: 일정 및 준비사항 표시

3. 업무 관리 (TaskManagement)

* 업무 목록: 모든 업무를 테이블 형태로 확인
* 필터링: 담당자·상태·검색어 기준으로 목록 좁히기
* 업무 생성: [새 업무 추가] 클릭 → 제목·설명·담당자·마감일 입력 → 저장
* 업무 편집: 기존 업무 클릭 → 내용 수정 후 저장
* 상태 변경: 대기중, 진행중, 완료 상태 변경
* 캘린더 뷰: 월별 업무 일정 확인
* 일별 상세: 특정 날짜의 업무 상세 보기

4. 통합 (Integration)

* Slack 연동
  + [Slack 연동] 클릭 → OAuth 로그인 → 워크스페이스 연결 → 상태 확인 / 해지 가능
* Notion 연동
  + [Notion 연동] 클릭 → OAuth 로그인 → 워크스페이스 연결 → 상태 확인 / 해지 가능
* Jira 연동:
  + [Jira 연동] 클릭 → OAuth 로그인 → 프로젝트 연결 → 상태 확인 / 해지 가능
* 연동 관리: 서비스별 설정 변경, 수동 동기화 실행

### 5. 설정 (Settings)

* 팀 관리
  + 팀원 추가·편집·삭제 가
  + 프로젝트 오너·관리자·팀원 중 역할 지정 가능
  + 부서별 팀원 목록 필터 적용
  + 페이지당 최대 6명 표시
* 알림 설정: 푸시, 회의, 마감일 알림 On/Off 선택
* 테마 설정: 라이트 / 다크 / 자동 모드 중 하나로 변경 가능

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 5. LLM 연동

#### 5.1 연동 목적

* LLM 모델 (Qwen3) 와 노션,지라 및 웹 애플리케이션 연동
* 회의 분석 및 프로젝트 기획안, 파생 업무 자동 생성

5.2 웹 대시보드 출력

1) 데이터 변환 과정

* LLM의 JSON 응답이 백엔드 서버에 도착하면, 먼저 PostgreSQL 데이터베이스에 저장
* 각 필드는 관계형 테이블 구조에 맞게 분리되어 저장

2) 실시간 표시

* WebSocket을 통해 프론트엔드로 즉시 전송
* React 컴포넌트가 데이터를 받아 프로젝트 카드 생성
* 프로젝트명은 카드 제목으로, 기간은 진행률 바로 표시
* 핵심 목표는 체크리스트 형태로 렌더링
* 실행 계획은 간트 차트와 타임라인 뷰로 시각화

5.3 Notion 출력

1) 페이지 생성

* Notion API를 호출하여 새로운 프로젝트 페이지를 생성
* LLM 응답의 각 필드가 Notion 블록으로 변환

2) 구조 매핑

* project\_name → 페이지 제목
* Project\_purpose → 첫 번째 단락 텍스트
* Core\_objectives → 번호 매긴 리스트
* Execution\_plan → 테이블 블록 (4개 컬럼: 단계, 기간, 목표, 방법)
* Expected\_effects → 불릿 포인트 리스트
* 태그 : "AI생성", "회의록 기반" 자동 추가

생성된 페이지는 팀 워크스페이스에 자동으로 공유되며, URL이 생성되어 슬랙과 대시보드에서 바로 접근가능

5.4 Jira 출력

1) 이슈 생성 프로세스

* LLM 응답을 Jira의 계층적 이슈 구조로 변환

2) 에픽 생성

* 전체 프로젝트가 하나의 에픽(Epic)으로 생성
* 제목 : project\_name 값 사용
* 설명 : project\_purpose와 expected\_effects 결합
* 기한 : project\_period에서 종료일 추출

3) 스토리 생성

execution\_plan 배열의 각 항목이 개별 스토리(Story) 취급

* 각 "단계"가 스토리 제목
* "목표"와 "방법"이 스토리 설명
* "기간"에서 스토리별 시작일과 종료일 설정
* 에픽과 자동 연결 (Epic Link 설정)

4) 우선순위 자동 할당

* AI가 판단한 중요도에 따라 Priority 필드가 자동 설정

5) 담당자 매핑

* project\_manager 필드의 이름을 Jira 사용자 ID와 매칭하여 자동으로 담당자를 할당

5.5. 데이터 동기화 메커니즘

- 일관성 유지

* 세 플랫폼 모두에 동일한 프로젝트 ID를 메타데이터로 저장하여, 한 곳에서 수정이 발생하면 다른 플랫폼도 업데이트

- 변경 추적:

* 대시보드에서 상태 변경 → Jira 이슈 상태 동기화
* Notion에서 마감일 수정 → Jira Due Date 업데이트
* Jira에서 담당자 변경 → 대시보드와 Notion에 반영

## 6. 보안 및 예외 처리

### 1) 오류 유형별 처리

* 인증 오류 (401): API 키 문제
* 타임아웃: 재시도 권장
* 서버 과부하 (503): 5초 후 재시도
* JSON 파싱 오류: 자동 재시도

### 2) 보안 기능

* API 키 암호화: 환경 변수에서 암호화된 키 관리
* Rate Limiting: 사용자당 분당 10회 요청 제한
* 키 복호화: crypto 모듈을 통한 안전한 키 사용

## 7. 모니터링 및 성능

### 1) 사용량 로깅

* 사용자별 토큰 사용량 추적
* 응답 시간 및 성공률 모니터링
* 일일 사용량 통계

### 2) 성능 최적화

* Redis 캐싱: 유사 회의록 결과 1시간 캐싱
* 해시 기반: 동일 내용 식별을 통한 중복 처리 방지
* 메트릭 수집: 평균 응답시간, 성공률, 오류율 추적

## 8. 프로덕션 고려사항

### 1) 안정성

* 30초 타임아웃 설정으로 무한 대기 방지
* 필수 필드 검증을 통한 데이터 무결성 보장
* 다단계 오류 처리로 서비스 안정성 확보

### 2) 확장성

* 모듈화된 서비스 구조로 유지보수 용이
* 캐싱 전략으로 서버 부하 최소화
* 실시간 모니터링으로 성능 이슈 사전 감지