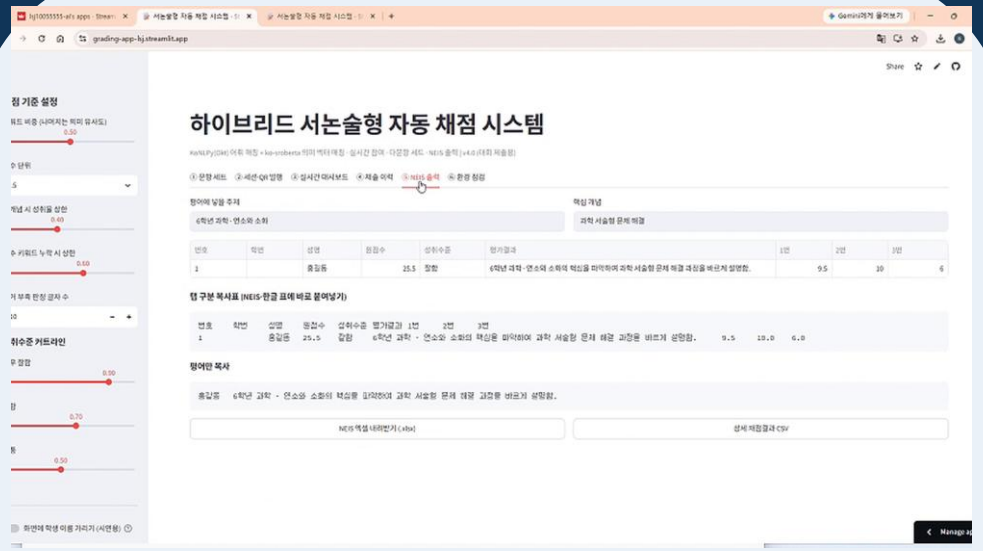
현장 피드백을 반영한

PNU-AIE
서·논술형 평가 지원 프로그램

교사 주도형 평가 설계 × AI 분석 보조



시연 사이트 QR링크

의미 기반 분석

하이브리드 안정성

가명화 관리

현장의 질문이 이번 개선의 출발점입니다.

지도교수 피드백

**키워드(규칙 기반) 방식 채점은
부정어 처리와 유의어 처리에 한계가 있다.**

→ 코사인 유사도 방식과 API 활용 요청

참여교사 피드백

현장에서 바로 쓰려면 무엇이 더 필요할까?

출석·미제출 확인 · 나이스 파일 출력 · 인터넷 장애 대응
예시 파일 · 학생 정보 가명화 보관

피드백을 크게 나누면

1 **정확성** 의미 중심 비교

2 **안정성** 하이브리드 폴백

3 **업무연계** 나이스·출결 흐름

4 **안전성** 가명화·보관 원칙

5 **사용성** 예시 파일·가이드

규칙 기반 채점과 코사인 유사도 기반 채점의 차이

PNU-AIE | 피드백 반영

같은 의미의 다양한 표현을 어떻게 해석할 것인가?

기존 방식 | 규칙 기반 인공지능(키워드 중심)

질문: "지정한 핵심 단어가 포함되었는가?"

판단 단위 특정 단어·표현의 포함 여부와 사전 설정 규칙

예시

모범: "빛에너지가 화학에너지로 전환된다."
학생: "식물은 빛을 이용해 영양분을 만든다."(유의)

모범: "독도는 대한민국의 영토이다."
학생: "독도는 일본의 영토가 아니다."(부정)

→ **표현이 달라지면 키워드 불일치로 판단될 수 있음**

장점

빠른 판정
명확한 규칙 설정

유의점

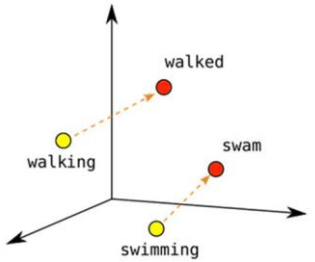
유의어·문맥 변화
부정 표현은 별도 규칙 필요

핵심: "단어가 있는가?"를 중심으로 확인

개선 방식 | 딥러닝 (코사인 유사도 활용 문장 의미 중심)

질문: "모범답안과 학생답안의 의미가 얼마나 가까운가?"

1 문장을 벡터로 변환



2 벡터 간 각도·방향을 비교



결과: 의미가 가까운 답안은 높은 유사도, 의미가 다르면 낮은 유사도 경향

유의: 부정어·예외·경계점 답안은 교사가 반드시 재검토

PNU-AIE 적용 방향: 문장 임베딩 기반 코사인 유사도 + 핵심 개념어 확인으로 1차 분석을 보조하고, 최종 판단은 교사가 수행합니다.

지도교수 피드백 반영

키워드 일치에서 의미 유사도 비교로 보완합니다.

기존: 규칙·키워드 중심 판정

판정 근거
지정 단어가 답안에 포함되었는지 확인

주의할 점
유의어·문장 구조·부정 표현의 차이를
충분히 반영하기 어렵습니다.

개선: 문장 의미 + 핵심 개념어 확인

문장의 의미 확인
모범답안과 학생 답안의 문장 벡터를 비교해
코사인 유사도를 산출





Hugging Face API 활용

보완 장치(핵심 개념어)
형태소 분석 기반 핵심 개념어를 함께 확인



Java Okt 형태소 분석기 활용

교사 검토 구역

창의적 표현, 경계에 해당하는 점수는 반드시 교사가 재검토합니다.

인터넷·모델 장애에 대비한 3단계 분석 구조입니다.

1순위

Hugging Face API

ko-sroberta 기반
의미 유사도 분석

2순위

로컬 Transformer

로컬 컴퓨팅 자원을
활용한 분석

3순위

TF-IDF 문자 n-gram

네트워크·모델 미작동 시
오프라인 폴백

운영 유의점 : 웹 배포판은 접속을 위해 인터넷이 필요합니다. 완전한 오프라인 운영은 설치 파일 실행을 준비해야 합니다.

배포 계획

GitHub 기반 이중 배포 계획

동일한 소스코드를 GitHub에서 관리하고, 교사의 사용 환경에 맞춰 웹과 실행파일로 제공할 계획입니다.

공통 기반

GitHub Repository

소스코드 · 버전관리 · Release 관리

WEB

Streamlit 웹 배포

GitHub push → Streamlit Cloud 배포 → 웹 브라우저 접속

GitHub › Streamlit Cloud › 웹 브라우저

- 별도 설치 없이 URL로 즉시 접속
- 코드 변경 시 최신 버전을 신속히 반영

LOCAL

실행파일 배포

GitHub Release → 실행파일 다운로드 → 개인 PC에서 실행

app.py
PNU-AIE 소스코드 및 실행파일.zip
README.md
requirements.txt

GitHub Release › 실행파일 › 개인 PC

- 교내·개인 PC 환경에 맞춘 로컬 실행 선택지
- 로컬 모델 사전 준비 시 네트워크 의존도 완화

운영 방식

하나의 코드 · 두 가지 이용 경로

GitHub에는 소스·문서·예시 파일만 관리하고, API 토큰과 실제 학생 개인정보는 별도 보관합니다.

※ 실행파일의 완전 오프라인 의미 분석은 로컬 임베딩 모델 사전 준비 여부에 따라 달라집니다.

UptimeRobot 정기 모니터링

운영 안정성

앱 URL을 5분 간격으로 확인하여 휴면 전환 가능성을 낮추고, 시연·수업 전 접속 지연을 줄입니다.

운영 방식

- **정기 HTTP 상태 확인**

5분 간격으로 앱 URL 점검

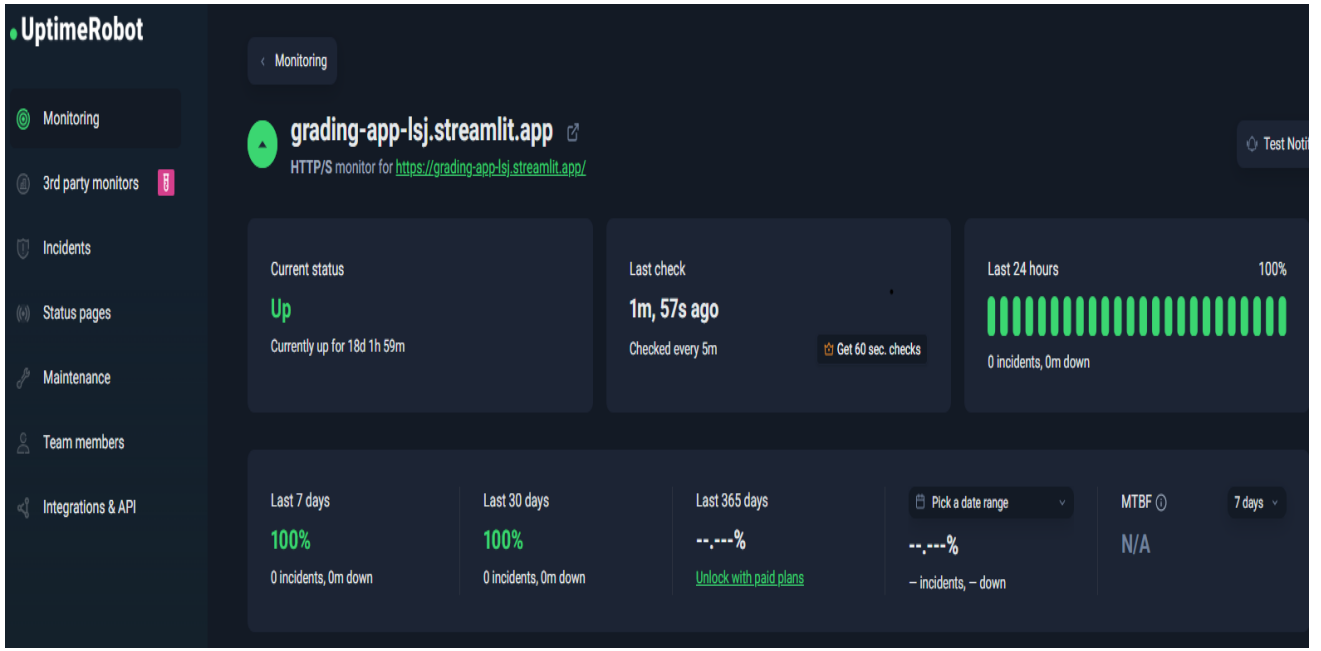
UptimeRobot 정기 호출 → 앱 URL 상태 확인 → Up

기대 효과

- 휴면 전환 가능성 및 첫 접속 지연 완화
- 시연·수업 전 서비스 상태 사전 확인
- 운영 중 장애 징후를 조기에 파악

UptimeRobot 모니터링 화면

설정 화면 예시



- 운영 원칙: UptimeRobot은 앱 URL을 주기적으로 확인하는 보조 수단입니다. 플랫폼 정책, 외부 API 장애 및 네트워크 오류까지 보장하지 않으므로 3단계 폴백 구조와 함께 운영합니다.

결과 파일은 선생님들의 의견이 더 필요합니다.

확인된 출력 기능

CSV / XLSX(나이스 업로드용)

포함 정보를 무엇으로 할지?
- 유사도 점수
- 최종 배점
- 세특 양식?

※ 실제 나이스 업로드 파일이 학교 급과 과목 별로 달라 의견이 필요합니다.

나이스 업로드 전 결과 파일 예시

학생코드	문항	유사도	최종배점	가명화	확인
학생01	1	0.82	8	예	교사 확인
학생02	1	0.54	5	예	경계 사례
학생03	1	—	—	예	미제출

시범 운영안

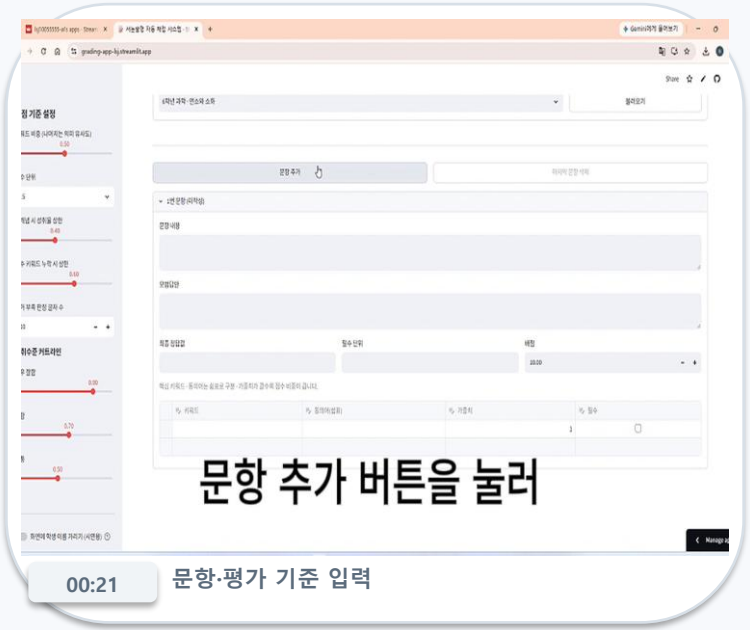
출석·결석·미제출 상태 및 초등 나이스 업로드 버전으로 준비.

※ 프로그램의 자동 처리 범위는 시범 적용에서 확인 후 확정합니다.

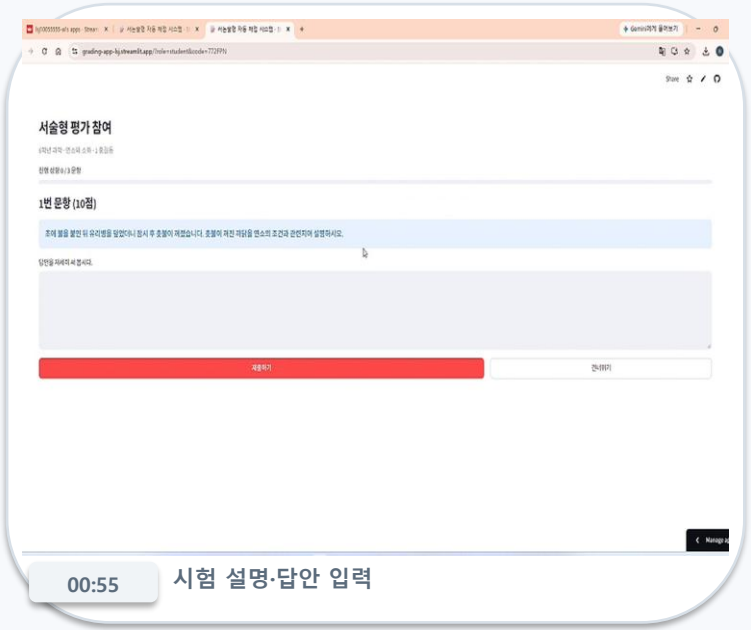
피드백 반영 현황: 기능·운영·시범 적용을 구분합니다.

피드백	반영 방식	상태
부정어·유의어 한계	의미 유사도 + 핵심 개념어 확인	기능 반영
나이스 입력 파일	CSV / XLSX(나이스 업로드용) 결과	기능 확인
인터넷 장애	API → 로컬 모델 → TF-IDF 폴백	기능 확인
예시 파일 필요	입력·결과·1쪽 사용 가이드 배포	운영 반영
학생 정보 보호	학생01 등 가명화 + 대응표 분리 보관	기능·운영
출석·미제출 확인	입력 양식·QR 출석/제출 확인표로 시범 적용	시범 적용

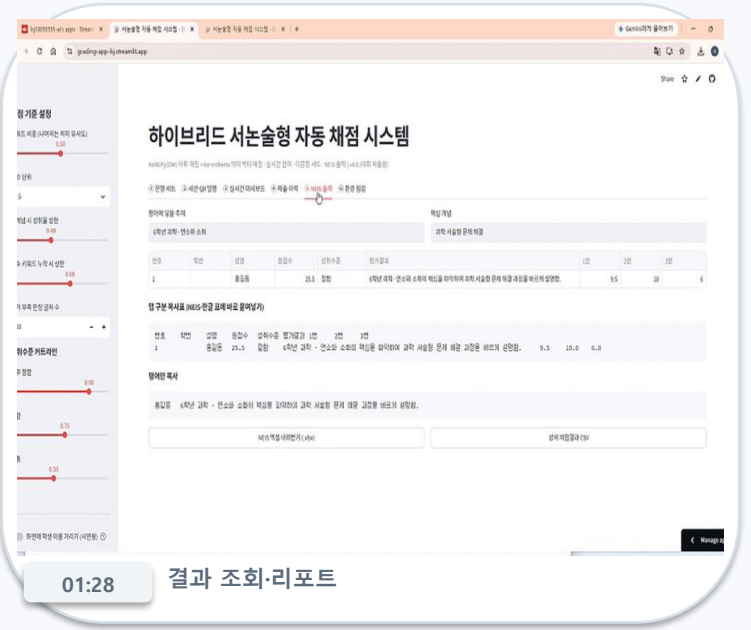
첨부 시연 동영상



00:21 문항·평가 기준 입력



00:55 시험 설명·답안 입력



01:28 결과 조회·리포트

시연 중 확인할 질문

1. 모범답안과 핵심 기준은 충분히 구체적인가?
2. 경계 사례를 교사가 어떻게 확인할 것인가?
3. 나이스 출력에 어떤 항목이 필요한가?

AI가 보조할 일과 교사가 판단할 일을 분명히 합니다.

AI 분석 보조

반복적 분석과 정리

- 모범답안–학생답안 유사도 비교
- 핵심 개념어 확인과 1차 점수 후보
- 유사도·정답률 분포 확인
- 결과 표와 CSV/XLSX 파일 생성

검토

교사 최종 판단

맥락·공정성·피드백

- 부정어·반대 주장·창의적 표현 확인
- 경계 점수와 예외 답안 재검토
- 최종 점수와 성적 처리 결정
- 학생에게 제공할 피드백 문구 결정

AI 결과는 최종 성적이 아니라 교사가 검토할 분석 자료입니다.