

AI 기반 발표 자동 평가 시스템

(Automated Presentation Evaluation System)

본 보고서를 Project 프로젝트의 최종보고서로 제출합니다.

학과(전공): 컴퓨터소프트웨어공학과

담당 교수: 권오준 교수님

팀 명: 무박2일

제출 일: 2025년 12월 05일

팀장: 임진호 (20212979)

팀원1: 이규찬 (20212977)

팀원2: 임민욱 (20213034)

팀원3: 이지민 (20233065)

동의대학교
컴퓨터소프트웨어공학과(전공)장 귀하

〈 요 약 문 〉

프로젝트 목표 (300자내외)	<p>본 프로젝트는 Overnight.AI라는 서비스명을 통해, 밤새 발표를 준비하는 학생과 직장인을 위한 AI 멘토를 제공하는 것을 목표로 한다. 기존 STT 중심의 평가 한계를 넘어, 컴퓨터 비전 기술을 접목한 멀티모달 발표 평가 시스템을 구축한다. 이를 통해 발표자의 언어적 요소뿐만 아니라 비언어적 요소를 통합 분석하여, 정량적이고 객관적인 평가 점수와 맞춤형 피드백을 제공함으로써 사용자의 발표 역량을 획기적으로 향상시키고자 한다.</p>		
내용 (500자내외)	<p>Android(Kotlin) 기반의 사용자 앱과 FastAPI(Python) 기반의 고성능 AI 분석 서버를 구축하였다. 사용자는 앱을 통해 발표 영상을 업로드하며, 맞춤형 채점 기준(평가 항목, 배점)을 설정할 수 있다.</p> <p>서버에서는 FFmpeg로 미디어를 분리한 후, OpenAI Whisper 모델로 음성을 텍스트로 변환한다. 비전 처리에는 Google MediaPipe를 활용하여 발표자의 478개 얼굴 랜드마크와 시선 벡터를 추적하며, Blendshapes를 통해 52가지 이상의 미세 표정 수치를 추출하여 웃음, 징그림, 시선 처리, 눈 깜빡임 등을 프레임 단위로 정량화한다. 또한 Praat 라이브러리로 목소리의 떨림과 거칠기를 분석한다.</p> <p>이렇게 추출된 멀티모달 데이터는 타임스탬프를 기준으로 융합되어 GPT-4o 기반의 채점 엔진에 입력되며, 최종적으로 항목별 점수와 상세 피드백이 담긴 JSON 결과가 생성된다. 결과물은 앱 내에서 직관적인 도넛 차트로 시각화되며, PDF 및 Excel 형식으로 자동 변환되어 사용자에게 제공된다.</p>		
기대효과 (200자내외)	<p>발표자는 시공간의 제약 없이 무제한으로 연습하고 객관적인 AI 피드백을 받아 스스로 약점을 보완할 수 있다. 평가자는 반복적인 채점 업무로 인한 피로도를 줄이고, 데이터에 기반한 공정한 평가를 수행할 수 있다. 궁극적으로 에듀테크 시장에서 경량화된 AI 채점 솔루션으로서 상용화 가능성을 확보한다.</p>		
Keywords	멀티모달 Whisper AI	컴퓨터 비전 MediaPipe	발표 자동 채점 생성형 AI

목 차

I. 서 론	4
1.1 프로젝트 배경 및 필요성	4
1.2 프로젝트 목표	5
II. 프로젝트 수행 내용	6
2.1 시스템 아키텍처 및 구성요소	6
2.2 상세 기술 스택 및 선정 이유	7
2.3 설계 제한요소 및 해결 방안	9
2.4 상세 구현 내용	10
2.5 시행착오 및 문제 해결	18
III. 결 론	19
3.1 프로젝트 결과 요약	19
3.2 목표 달성을 평가	19
3.3 향후 발전 방향	20
IV. 참고문헌	21
[부록]	22
[소감]	24

I. 서론

1.1 프로젝트 개요 및 배경

1) 프로젝트 대상의 소개

Overnight.AI는 발표 영상을 업로드하면 AI가 시각과 청각 데이터를 분석하여 점수와 피드백을 제공하는 자동 평가 시스템이다. 단순히 말을 텍스트로 바꾸는 것을 넘어, 발표자의 태도를 정량화하는 데 초점을 맞추었다.

2) 기존 문제점 분석

- 발표자 입장:** 연습 후 객관적인 피드백을 받기 어렵고, 자신이 어떤 부분(시선 처리, 자세 등)에서 부족한지 스스로 파악하기 힘들다.
- 평가자 입장:** 다수의 학생을 반복적으로 평가해야 하므로 피로도가 누적되며, 주관적인 판단 개입으로 인해 평가의 일관성을 유지하기 어렵다. 또한 결과 정리 및 피드백 작성에 많은 시간이 소요된다.

발표자 입장	평가자 입장
<p>발표 잘했는데 왜 점수가 낮지....</p> <p>🤔</p> <p>발표 피드백 해줄 사람?</p> <p> 발표 피드백 어디서 받아요</p> <p>😴 😤 😤</p>	<p>발표자 입장</p> <ul style="list-style-type: none">연습 후 객관적인 피드백을 받기 어려움발표 내용 중 부족한 부분을 명확히 파악하기 어려움
<p>팁별 점수는 언제 다 기입하지...</p>  <p>내가 일관성 있게 평가했나?</p> <p>참여자에게 피드백도 보내줘야 하는데...</p>	<p>평가자 입장</p> <ul style="list-style-type: none">반복적인 발표 채점은 시간이 오래 걸리고 피로도가 큼주관적인 판단으로 인해 점수 일관성 확보 어려움결과 정리 및 기록 또한 수작업으로 번거로움

3) 프로젝트 필요성 및 의도

최근 교육청 등에서 AI 기반 논술/평가 시스템 도입이 확대되고 있는 추세이다. 하지만 기존 상용 서비스들은 대부분 영어 발음 교정이나 텍스트 문법 분석에 치우쳐 있다. 우리 팀은 비언어적 요소까지 통합 평가할 수 있는 시스템이 필수적이라고 판단하였다. 이를 통해 교육 현장의 공정성을 확보하고 효율적인 발표 훈련 도구를 제공하고자 한다.

1.2 프로젝트 목표

Overnight.AI 프로젝트의 목표는 발표자의 언어적·비언어적 요소를 동시에 분석할 수 있는 AI 기반 멀티모달 발표 평가 시스템을 개발하는 데 있다. 기존 STT 중심의 평가 방식은 발화 내용에만 치우쳐 있어, 발표자의 태도나 비언어적 표현을 반영하기 어렵다는 한계가 존재한다. 본 프로젝트는 이러한 문제점을 해결하기 위해 영상 분석과 음성 분석을 융합하여, 보다 객관적이고 정량적인 평가를 제공한다.

본 시스템은 다음과 같은 핵심 목표를 바탕으로 설계되었다.

1. 정확성 향상

- Whisper, MediaPipe 등 검증된 AI 모델을 활용하여 음성 인식률 및 얼굴 랜드마크 추적 정확도를 확보한다.
- 오디오·비디오 데이터를 타임스탬프 기준으로 융합하는 알고리즘을 적용한다.

2. 객관성 및 신뢰성 확보

- GPT 기반 채점 엔진을 통해 일관된 평가 기준으로 점수와 피드백을 생성하여 평가자의 주관 개입을 최소화한다.

3. 사용자 편의성 강화

- Android 앱을 통해 영상 업로드 → 분석 → 결과 확인의 단순한 사용자 흐름을 제공한다.
- 분석 결과를 PDF 리포트로 자동 생성하여 교수·평가자에게 즉시 제출 가능하도록 지원한다.

4. 확장성 및 실서비스 가능성 확보

- FastAPI 기반 비동기 서버 구조를 통해 대용량 영상 처리 성능을 확보하고, 향후 평가 기준·피드백 확장 및 LMS 연동 등 상용화 단계로 확장할 수 있도록 설계한다.
- 이를 통해 Overnight.AI는 발표 연습의 접근성과 효율을 향상시키고, 학습자의 역량 향상을 위한 개인 맞춤형 AI 멘토 서비스로 발전하는 것을 목표로 한다.

II. 프로젝트 수행 내용

2.1 시스템 아키텍처 및 구성요소

본 프로젝트는 **Android-Client**, **FastAPI-Server**, **Firebase-DB**의 3계층 아키텍처로 설계되었다.

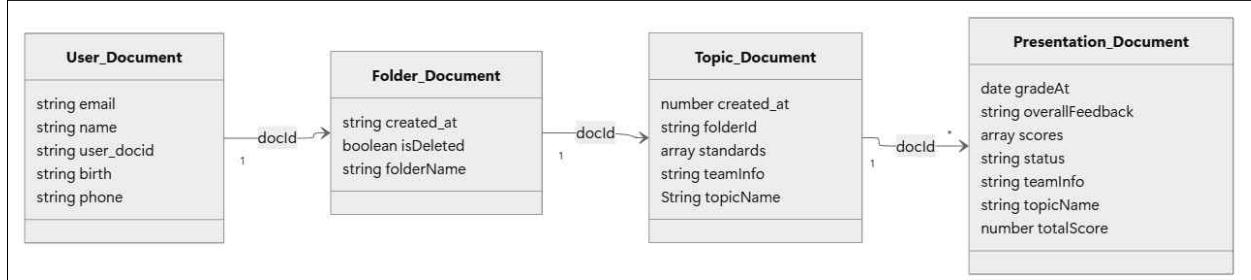
설계 구성요소	내용
합성	<ul style="list-style-type: none">주제 선정: 음성(STT)과 영상(Vision)을 융합한 멀티모달 발표 평가 시스템요구 분석: 발표자에게는 상세 피드백을, 평가자에게는 자동화된 리포트를 제공하는 기능 설계역할 분담: 프론트엔드와 백엔드로 나누어 협업 진행일정 수립: 15주차에 걸친 단계별 개발 계획 수립
분석	<ul style="list-style-type: none">요소 파악: 고성능 STT를 위한 Whisper, 실시간 얼굴 인식을 위한 MediaPipe, 추론을 위한 GPT-4 선정기능 이해: 각 AI 모델의 입출력 데이터 포맷 및 처리 지연 시간 분석방법 마련: 대용량 영상 처리를 위해 FastAPI의 비동기 처리 방식 채택상호 작용: 오디오의 시간축과 비디오의 프레임축 데이터를 동기화 하는 알고리즘 설계
제작	<ul style="list-style-type: none">각 요소들의 구현: Kotlin으로 안드로이드 앱 구현, Python FastAPI로 AI 분석 서버 구현각 요소들을 통합한 전체 시스템의 구현: Retrofit을 이용한 REST API 통신을 통해 업로드–분석–결과 반환까지의 엔드투엔드 파이프라인을 완성시각화 및 리포트: MultiSegmentDonutChart 커스텀 뷰를 구현하여 평가 기준별 점수를 도넛 차트로 표현하고, 결과 데이터는 PDF 및 Excel 리포트로 저장 할 수 있도록 기능을 구성
시험	<ul style="list-style-type: none">단위 테스트: 각 AI 모델별 인식률 테스트 및 API 엔드포인트 응답 속도 측정통합 테스트: 실제 발표 영상을 업로드하여 PDF 리포트가 생성되기 까지의 전 과정 검증디버깅: 대용량 파일 업로드 시 타임아웃 문제 및 한글 폰트 깨짐 현상 수정
평가	<ul style="list-style-type: none">충족 여부: 계획했던 멀티모달 분석 및 자동 리포트 생성 기능 구현 완료정량화: 해외 연구 사례를 참고하여 Whisper의 신뢰성을 확보하고, 자체 테스트 결과 만족스러운 인식률 확인보완: 사용자 피드백을 반영하여 UI의 직관성을 높이고 로딩 화면에 진행률 표시 기능 추가

기타	<ul style="list-style-type: none"> - 도구 활용: Git(형상관리), Notion(문서화), Discord(커뮤니케이션) 적극 활용 - 참고 자료: 최신 AI 논문 및 기술 블로그, 유사 서비스 벤치마킹
----	--

2.2 상세 기술 스택 및 선정 이유

1. **Kotlin (Android Client)**: JetBrains에서 개발한 현대적인 언어로, Java와 100% 호환되며 비동기 처리가 강력하여 네트워크 통신이 잦은 본 프로젝트에 적합하다.
2. **FastAPI (Backend Server)**: Python 기반의 고성능 웹 프레임워크로, 비동기 처리를 기본 지원하여 무거운 AI 모델 서빙에 최적화되어 있다. 자동 문서화 기능을 통해 프론트엔드와의 협업 효율을 높였다.
3. **Firebase Firestore (Cloud NoSQL DB)**: 문서 기반의 NoSQL 데이터베이스로 계층적 구조 (User → Folder → Topic → Presentation)에 유리하다. 스키마 변경이 유연하며 모바일 환경에서의 빠른 읽기 성능을 제공한다.
4. **Firebase Authentication (Account & Security)**: 이메일/비밀번호 방식 회원가입과 Google 계정 소셜 로그인을 지원한다. 또한 sendPasswordResetEmail()을 통해 모든 외부 이메일 (Naver 포함)에 비밀번호 재설정 링크를 발송한다.
5. **Whisper AI (STT)**: OpenAI의 고성능 음성 인식 모델로, 소음 환경에서도 강하며 한국어 인식률이 매우 뛰어나다. Doshisha 대학 연구 결과 등을 통해 그 신뢰성이 입증되었다.
6. **MediaPipe (Computer Vision)**: Google의 온디바이스 머신러닝 솔루션으로, 478개의 얼굴 랜드마크와 손동작을 실시간으로 추적할 수 있어 발표 태도 분석에 핵심적인 역할을 수행한다.
7. **OpenAI (Reasoning)**: 수치화된 데이터를 바탕으로 자연스러운 문장의 피드백을 생성하고, 사용자가 설정한 채점 기준에 맞춰 논리적인 평가를 수행한다.
8. **iText + Apache POI (결과 리포트 생성)**: iText는 한글 폰트를 포함한 PDF 성적표 자동 생성에 사용되었으며, Apache POI(XSSFWorkbook)는 동일 주제 내 발표 점수를 비교하는 Excel 파일 생성에 사용되었다.

데이터베이스 ERD



user	
user	75hm6dHS94cdrxWYqdZeuv7TNm2
+ 컬렉션 시작	+ 문서 추가
user	75hm6dHS94cdrxWYqdZeuv7TNm2
	0e17w9zmonVzcMct74b0nF2ZCPQ2
	WCNA7UgPk1Sb1FbCfusci4m7txYZ
	WK3Ds5R7yNfh1B8opPxIOvoqwf1
	mWICm77aVmPuISxFbsP8VR3F25P2
	pCefqtl2MopCT3Fdai14nNwb4qbH2
	pGyKUccbgyUFu7zg1oyt1WhPA23
	tfr5xjbzzXw4SNnnIL2dZSquaJ2
	+ 컬렉션 시작
	folders
	+ 필드 추가
	birth: ""
	email: "leekyuchan6079@gmail.com"
	name: "이규찬"
	phone: ""
	user_docid: "75hm6dHS94cdrxWYqdZeuv7TNm2"

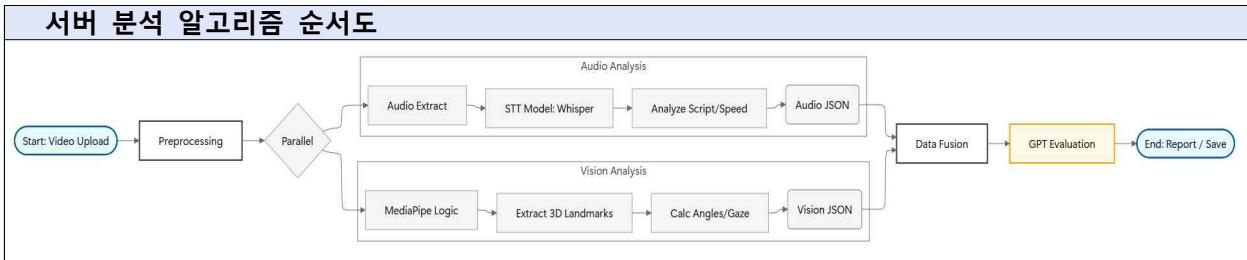
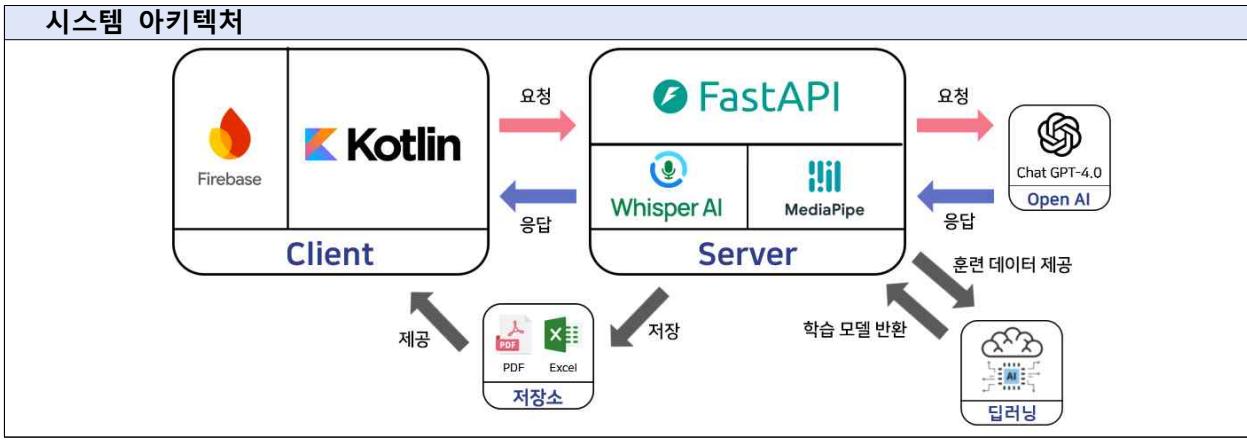
folders		Google Cloud의 추가 기능	
user > 75hm6dHS94cdrhxWYqdZeuw7TNm2...			
75hm6dHS94cdrhxWYqdZeuw7TNm2	⋮	folders	⋮
+ 컬렉션 시작		+ 문서 추가	
folders	>	2Dj8ZpU49rMuw4gteVi3	>
		XTZbQ49IIsqZACTirg4	
		hP9Oqkz2LmDoNY9RkL7N	
+ 필드 추가		+ 컬렉션 시작	
birth: ""		topics	
email: "leekyuchan6079@gmail.com"			
name: "이규찬"			
phone: ""			
user_docid: "75hm6dHS94cdrhxWYqdZeuw7TNm2"			
+ 필드 추가		+ 필드 추가	
		created_at: 1763968641355	
		isDeleted: false	
		name: "capstone design"	

topics

▶ user > 75hm6dHS94cd... > folders > 2Dj8ZpU49rMu... > topics > AGoLa79sk1qpenydr4i

Google Cloud의 추가 기능

2Dj8ZpU49rMuw4gteV3	topics	AGoLa79sk1qpenydr4i
+ 컬렉션 시작 topics	+ 문서 추가 4f6z2ZrxxF1QxE5fM3nZ : AGoLa79sk1qpenydr4ii MSFtq4B9iXRbmY2PgEvd e6s5SwHe16XL9k77z76A hz7k1i9QdgbsswC9cY0 iHpnksSp7zw2OBymxI49 oML2ndERLgnlpw87PRB0	+ 컬렉션 시작 presentations
+ 필드 추가 created_at: 1763968641355 isDeleted: false name: "capstone design"	+ 필드 추가 createdAt: 2025년 11월 24일 PM 8시 29분 16초 UTC+9 folderId: "2Dj8ZpU49rMuw4gteV3" - standards - 0 standardDetail: "Are you staring straight ahead" standardName: "Gaze processing" standardScore: 50 - 1 standardDetail: "Did you introduce yourself before the presentation" standardName: "content" standardScore: 50 teamInfo: "overnightABCD" topicName: "final presentation"	



2.3 설계 제한요소 및 해결 방안

설계 제한요소	내용
원가	<ul style="list-style-type: none"> 비용 절감: 상용 API 사용을 최소화하기 위해 오픈소스 모델을 로컬 서버에 직접 호스팅하여 비용을 절감함
안정성	<ul style="list-style-type: none"> 예외 처리: 대용량 영상 업로드 시 네트워크 불안정으로 인한 앱 종료를 방지하기 위해 재시도 로직 및 예외 처리 구문 적용 비동기 처리: 분석 시간이 길어질 수 있으므로 서버는 비동기 작업 큐를 사용하고, 앱은 Polling 방식으로 진행 상황을 체크하여 앱 멈춤 현상 방지
신뢰성	<ul style="list-style-type: none"> 데이터 일관성: 동일한 영상에 대해 매번 유사한 점수가 나오도록 프롬프트 엔지니어링을 최적화함. 오류 최소화: 조명 변화나 얼굴 각도에 따른 인식 실패를 보정하기 위해 전후 프레임 보간법 적용.
미학	<ul style="list-style-type: none"> UI/UX: 사용자가 복잡한 기술을 몰라도 쉽게 사용할 수 있도록 촬영한 영상 업로드 -> 결과 확인의 2단계 심플한 흐름으로 구성. 시각화: 긍정/부정 피드백을 색상으로 구분하고, 도넛 차트를 활용해 직관적인 점수 확인 가능.
내구성	<ul style="list-style-type: none"> 리소스 관리: 서버의 메모리 누수를 방지하기 위해 분석이 완료된 임시 파일을 즉시 삭제하는 Cleanup 로직 구현. 환경 대응: 다양한 안드로이드 화면 해상도에 대응하는 반응형 레이아웃 적용.

2.4 상세 구현 내용

1) 발표 정보 및 채점 기준 설정

사용자는 앱에서 발표 주제와 팀 정보를 입력하고, 이번 발표에서 중점적으로 평가받고 싶은 항목(예: 전달력 40%, 태도 30%, 내용 30%)을 설정한다. 이 데이터는 Firebase Firestore에 저장된다.

2) 영상 업로드 및 전처리

사용자가 촬영한 영상은 Retrofit을 통해 서버로 전송된다. 서버는 FFmpeg를 사용하여 영상을 오디오(.wav)와 프레임 이미지(jpg)로 분리한다.

3) 멀티모달 AI 분석 (병렬 처리)

Vision: face_analyzer.py 모듈이 MediaPipe를 로드하여 각 프레임의 시선 처리, 표정 변화, 자세 안정성을 수치화한다.

Audio: audio_analyzer.py 모듈이 Whisper를 통해 텍스트와 타임스탬프를 추출하고, Praat 라이브러리로 목소리의 떨림을 분석한다.

4) 데이터 융합 및 점수 산출

서로 다른 주기를 가진 영상과 음성 데이터를 타임스탬프 기준으로 정렬한다. 융합된 데이터와 사용자의 채점 기준을 GPT 프롬프트에 입력하여 최종 JSON 리포트를 생성한다.

[자료 삽입 4] 서버 분석 로직 흐름도 (PPT 슬라이드 11 활용 가능) (Input -> Processing(Vision/Audio) -> Fusion -> Output 흐름)

5) 결과 시각화 및 리포트 저장

분석 결과는 AnalysisResultActivity에서 커스텀 도넛 차트(MultiSegmentDonutChart) 와 기준별 리스트로 시각화된다. 사용자는 'PDF 저장' 버튼을 눌러 현재 발표의 점수와 피드백을 A4 형식의 PDF 리포트로 저장할 수 있다.

또한 'Excel 저장' 기능을 통해, 동일한 주제(Topic)에 속한 발표들 중 status = "completed" 상태의 데이터만 모아 Apache POI 기반의 .xlsx 파일로 내보낼 수 있다.

이 엑셀 파일의 첫 행에는 팀명 + 평가 기준 이름 + 총점이 헤더로 생성되고, 각 행마다 한 발표의 점수가 채워져 팀별 점수 비교표로 활용된다.

2.4.1 앱 사용자 흐름

사용자가 앱을 실행하여 회원가입부터 최종 분석 결과를 확인하기까지의 사용자 흐름은 다음과 같이 구현되었다.

1) 인증 및 메인 화면

- **회원가입/로그인:** 사용자는 이메일·비밀번호 기반의 자체 회원가입을 통해 계정을 생성할 수 있으며, Google 계정 연동을 통한 소셜 로그인도 지원한다. 비밀번호를 잊어버린 경우, 비밀번호 찾기 화면에서 가입 시 사용한 이메일(예: Naver, Gmail 등)을 입력하면 Firebase Auth의 sendPasswordResetEmail()을 통해 해당 메일 주소로 비밀번호 재설정 링크가 전송된다.
- **메인 홈:** 로그인 후 진입하는 첫 화면으로, 사용자는 다음 기능을 이용할 수 있다.
 - **Script(대본 보조):** 발표 대본 작성률 높기 위한 챗봇 UI를 구현하였다. 현재는 대화형 채팅 인터페이스, 메시지 리스트, 입력창 등 프론트엔드 화면과 상호작용 흐름까지 구현되었으며, 실제 GPT 백엔드 연동은 차기 과제로 남겨두었다. 프로토타입 단계에서는 더미 응답을 사용하여 대화 흐름을 시뮬레이션하였다.
 - **Guide(가이드):** 본 시스템에서 고득점을 받기 위한 팁과 앱 사용 방법을 안내한다.
 - **서비스 소개:** 화면을 스크롤하면 OvernightAI의 핵심 가치와 기능을 직관적인 카드 뉴스 형태로 확인할 수 있다.

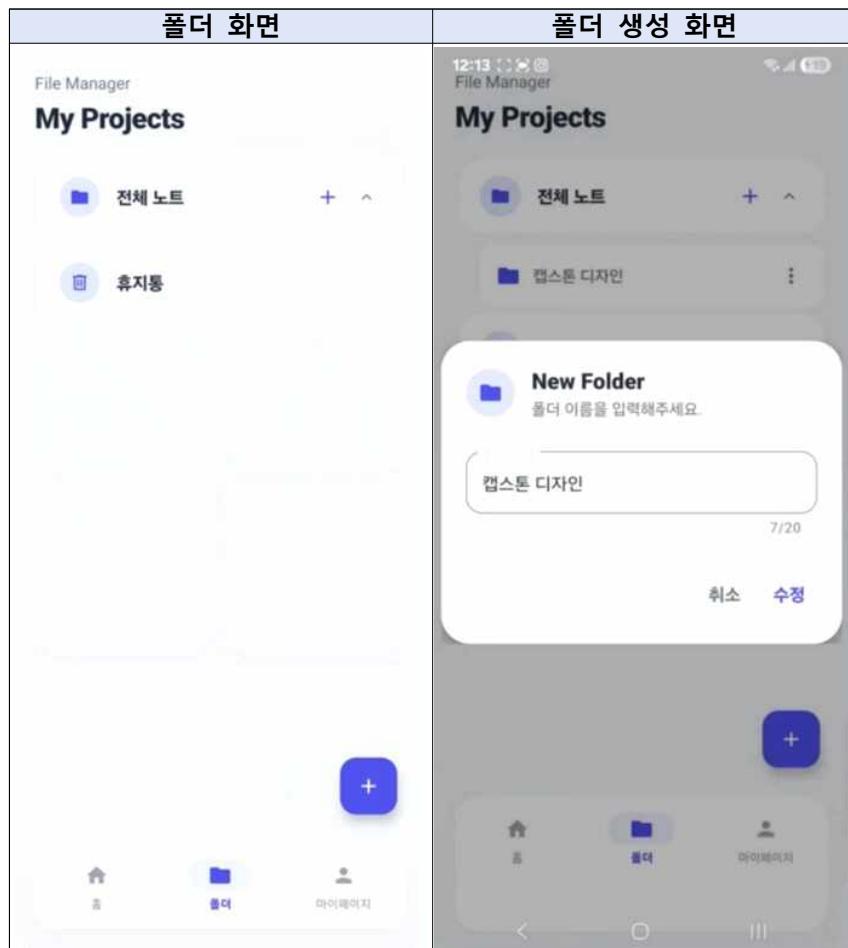
로그인 화면	회원가입 화면
<p>OvernightAI</p> <p>Welcome Back!</p> <p>이메일과 비밀번호를 입력하여 로그인하세요.</p> <p>Email <input type="text"/></p> <p>Password <input type="password"/> <input type="button" value="Eye icon"/></p> <p>비밀번호를 잊으셨나요?</p> <p>Log In</p> <p>Or continue with</p> <p> GOOGLE로 계속하기</p> <p> MICROSOFT로 계속하기</p> <p>계정이 없으신가요? 회원가입</p>	<p>← Create Account 회원가입을 진행해주세요.</p> <p>이름 <input type="text"/></p> <p>이메일 <input type="text"/></p> <p>생년월일 <input type="date"/></p> <p>휴대폰 번호 <input type="text"/></p> <p>비밀번호 <input type="password"/> <input type="button" value="Eye icon"/></p> <p>회원가입 완료</p> <p>이미 계정이 있으신가요? 로그인</p>

메인 홈 화면	서비스 소개 화면
<p>Overnight</p> <p>Hello, 임민국!</p> <p>Let's Practice</p>  <p>Script 대본 작성하기</p> <p>Guide 평가 기준 확인</p> <p>Tip. 자신감 있는 목소리는 높은 점수를 받습니다.</p> <p>서비스 소개 밤새 준비한 당신의 발표, AI가 정성껏 채점해드립니다. 발표 영상부터 점수 확인까지, 지금 바로 시작하세요.</p> <p>Script 대본 작성하기</p> <p>Guide 평가 기준 확인</p> <p>Tip. 자신감 있는 목소리는 높은 점수를 받습니다.</p> <p>서비스 소개 밤새 준비한 당신의 발표, AI가 정성껏 채점해드립니다. 발표 영상부터 점수 확인까지, 지금 바로 시작하세요.</p> <p>Home 폴더 마이페이지</p>	<p>Script 대본 작성하기</p> <p>Guide 평가 기준 확인</p> <p>Tip. 자신감 있는 목소리는 높은 점수를 받습니다.</p> <p>서비스 소개</p> <p>공정하고 일관된 평가 사전 기준에 따라 AI가 누구에게나 동일하게 채점하고 평가자의 주관 없이 신뢰도 높은 결과를 제공합니다.</p> <p>평가 시간 절약 반복되는 발표 평가를 자동화해 교사와 학생 모두 시간을 아낄 수 있어요.</p> <p>발표력 향상 발표 전 AI로 미리 점검하고 점수를 기반으로 개선 방향을 확인할 수 있어요.</p> <p>Home 폴더 마이페이지</p>

Script 화면	Guide – 평가 기준	Guide – 이용 방법
<p>←</p> <p>AI Script Coach</p> <p>안녕하세요, Presenter님! 오늘 발표 대본 작성 도와드릴까요?</p> <p>대본 작성은 도와드릴까요? ></p>	<p>← 가이드</p> <p>평가 기준 이용 방법</p> <p>AI는 발표를 어떻게 평가하나요?</p> <p>성공적인 발표를 위해 AI가 분석하는 3 가지 핵심 요소를 확인해보세요.</p> <p>전달력 (Delivery) 목소리의 크기, 빠르기, 발음의 정확도를 분석합니다. 너무 빠르거나 느리지 않은 적절한 속도와 자신감 있는 목소리 톤을 유지하세요.</p> <p>내용 구성 (Script) 주제와의 연관성과 논리적인 흐름을 평가합니다. '서론-본론-결론'의 구조가 명확하고 핵심 키워드가 적절히 포함되었는지 확인합니다.</p> <p>태도 (Attitude) 시선 처리와 자세를 분석합니다. 정면을 응시하며 청중과 교감하는 듯한 자연스러운 시선 처리는 높은 점수를 받습니다.</p>	<p>← 가이드</p> <p>평가 기준 이용 방법</p> <p>폴더 관리부터 결과 다운로드까지</p> <p>Overnight를 100% 활용하는 상세 단계를 확인해보세요.</p> <ol style="list-style-type: none"> 폴더 생성 및 대본 준비 <ul style="list-style-type: none"> '폴더' 탭에서 발표 주제별로 폴더를 미리 생성하세요. 대본이 막막하다면 흡 화면에서 AI에게 작성 요청할 수 있습니다. 발표 정보 및 기준 설정 <ul style="list-style-type: none"> 하단 '+' 버튼을 눌러 생성한 폴더를 선택하고 정보를 입력합니다. '항목 추가' 버튼으로 채점 기준을 최대 5개까지 상세하게 설정할 수 있습니다. 결과 분석 및 파일 저장 <ul style="list-style-type: none"> 영상은 업로드하면 점수, 피드백, 다시보기 영상이 제공됩니다. 분석 결과는 하단 버튼을 통해 Excel 또는 PDF로 다운로드할 수 있습니다.

2) 프로젝트 관리 및 발표 설정

- **폴더 관리:** 폴더 화면에서 '전체 노트' 하위에 과목별/주제별 폴더를 생성하여 발표 기록을 체계적으로 관리한다.
- **발표 정보 입력:** 하단 + 버튼을 눌러 '파일 업로드'를 선택하면 정보 입력 화면으로 전환된다. 여기서 사용자는 저장할 폴더를 지정하고, 팀 이름, 대회 및 수업 이름을 작성한다.
- **사용자 정의 평가 기준:** 사용자는 이번 발표에서 평가받고자 하는 항목(예: 전달력, 시선처리 등)과 내용, 배점을 직접 설정할 수 있다. 평가 항목은 최대 5개까지 유동적으로 추가 가능하다.



파일 업로드 버튼 화면	발표 정보 입력 화면	발표 항목 최대 5개
<p>File Manager</p> <h3>My Projects</h3> <p>전체 노트</p> <p>캡스톤 디자인</p> <p>휴지통</p> <p>파일 업로드</p> <p>폴더 추가</p> <p>+</p> <p>홈</p> <p>폴더</p> <p>마이페이지</p>	<p>←</p> <h3>발표 정보 입력</h3> <p>발표에 대한 정보를 입력해주세요</p> <p>캡스톤 디자인1</p> <p>팀 이름 무박2일</p> <p>대회 및 수업 이름 캡스톤 경진대회</p> <p>발표 항목 및 비중</p> <p>+ 항목 추가</p> <p>항목명 제스처</p> <p>내용 제스처를 활용하여 잘 표시하는지</p> <p>점수 50</p> <p>파일 업로드하기</p>	<p>←</p> <h3>발표 정보 입력</h3> <p>발표에 대한 정보를 입력해주세요</p> <p>항목명 평가 항목 4</p> <p>내용</p> <p>점수</p> <p>항목명 평가 항목 5</p> <p>내용</p> <p>점수</p> <p>항목은 최대 5개까지 추가할 수 있습니다.</p> <p>파일 업로드하기</p>

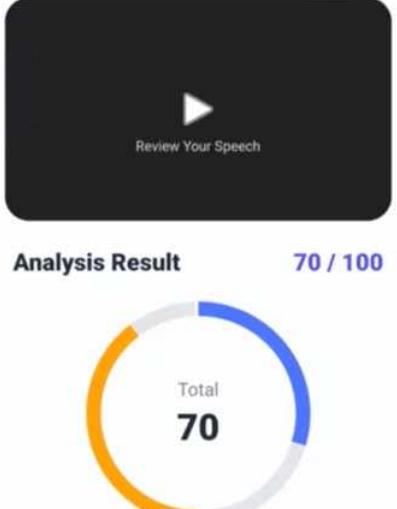
3) 영상 업로드 및 분석 요청

- 영상 선택:** 정보 입력 후 '파일 업로드'를 누르면 다운로드와 연동되어 사전에 촬영해 둔 발표 동영상을 선택할 수 있다
- AI 분석 시작:** 영상을 선택하고 'AI 분석 시작하기' 버튼을 누르면, 영상 데이터와 앞서 설정한 채점 기준이 서버로 전송된다. 서버에서는 FFmpeg, MediaPipe, Whisper가 순차적으로 동작하며 비디오/오디오/텍스트를 분석한다.

영상 선택 화면	분석 로딩 화면
<p>Upload Video</p> <p>발표 영상을 업로드하여 분석을 시작하세요.</p> <p>파일 변경하기 MP4, MOV format supported</p> <p>1000021474.mp4</p> <p>AI 분석 시작하기</p>	<p>3/6: 얼굴 데이터 분석 중...</p> <p>잠시만 기다려 주세요.</p> <p>6/6: 데이터 정렬 및 AI 채점 중...</p> <p>잠시만 기다려 주세요.</p>

4) 시각화된 평가 결과 확인

- 도넛 차트:** 분석 완료 시, 사용자가 설정한 평가 항목별 획득 점수가 도넛 차트로 시각화되어 목표 달성을 직관적으로 파악할 수 있다.
- 파일 저장:** 결과 화면 하단의 버튼을 통해 성적표를 PDF 문서나 Excel 파일로 즉시 다운로드하여, 한눈에 점수를 쉽게 확인할 수 있다.

결과 차트 화면	상세 피드백 화면
 <p>Analysis Result</p> <p>70 / 100</p> <p>Total 70</p> <p>발표는 전반적으로 친근한 분위기를 가지고 있었으나, 제스처와 목소리 크기에서 개선이 필요합니다. 제스처를 통해 메시지를 더욱 강조하고, 목소리의 일관성을 유지하는 연습을 하세요. 관객과의 시선 교환을 통해 소통을 강화하는 것도 좋은 방법입니다.</p>	<p>[영상 요약]</p> <p>발표자는 테스트를 위해 영상을 촬영하고 있으며, 친구들에게 가벼운 농담을 던지고 있습니다. 분위기는 편안하지만, 메시지를 전달하는 데 있어 좀 더 적극적인 제스처가 필요합니다.</p> <p>Details</p> <p>제스처</p> <p>발표 중 제스처의 활용이 부족하여 청중의 주목을 끌기 어려웠습니다. 더 많은 손동작이나 몸짓을 사용하여 메시지를 강조하는 것이 좋습니다.</p> <p>획득 점수: 30 / 50</p> <p>목소리 크기</p> <p>목소리 크기는 전반적으로 괜찮았으나, 몇몇 구간에서 다소 작게 들렸습니다. 일관된 목소리 크기를 유지하는 연습이 필요합니다.</p> <p>획득 점수: 40 / 50</p> <p>EXCEL PDF</p>

실제 파일(EXCEL)	실제 파일(PDF)										
 <table border="1"> <thead> <tr> <th>팀명</th> <th>발음</th> <th>시선처리</th> <th>자기소개</th> <th>총점</th> </tr> </thead> <tbody> <tr> <td>무박2일</td> <td>25</td> <td>10</td> <td>40</td> <td>75</td> </tr> </tbody> </table>	팀명	발음	시선처리	자기소개	총점	무박2일	25	10	40	75	<p>무박2일_Result.pdf</p> <p>무박2일 팀 분석 결과</p> <p>■ 평가 기준</p> <ul style="list-style-type: none"> • 시선처리 (30점) • 발음 (30점) • 자기소개 (40점) - 총 배점 : 100점 <p>■ 상세 피드백</p> <p>▶ 시선처리 점수: 10 / 30 피드백: 발표 중 정면을 응시하지 않고, 다른 방향으로 시선을 두는 경우가 많았습니다.</p> <p>▶ 발음 점수: 25 / 30 피드백: 발표 이 전반적으로 괜찮으나, 몇몇 단어에서 명확성이 부족했습니다.</p> <p>▶ 자기소개 점수: 40 / 40 피드백: 발표 시작 전에 자기소개를 잘 하였습니다.</p> <p>■ AI 종합 평가</p>
팀명	발음	시선처리	자기소개	총점							
무박2일	25	10	40	75							

5) 마이페이지

- 활동 내역 통계: 사용자의 누적 활동 데이터를 시각화하여 보여주는 UI를 설계하였다. 현재는 총 발표 횟수, 등록된 과목, 우수 발표(A등급) 등의 지표를 카드 형태로 표시하는 화면 구성을 완료하였으며, 실제 분석 결과 데이터와의 연동 및 통계 로직은 차기 과제로 남겨두었다.
- 계정 관리: 로그아웃 기능은 Firebase Auth와 연동하여 정상 동작한다.



2.5 시행착오 및 해결

1. 데이터베이스 모델링 구조 변경 (RDBMS → NoSQL)

- 문제점: 초기 데이터베이스 설계 시, 일반적인 관계형 데이터베이스(RDBMS)의 테이블 참조 방식을 적용하였다. 그러나 프로젝트에 도입한 Firebase Firestore는 NoSQL 기반의 문서 중심 데이터베이스로, 테이블 조인보다는 데이터의 계층적 구조화가 중요하다는 차이점이 있었다. 이로 인해 초기 설계대로 구현 시 데이터 쿼리가 복잡해지고 성능이 저하되는 문제가 발생했다.
- 해결 방안: Firestore의 특성에 맞춰 데이터베이스 스키마를 전면 재설계하였다. 참조 방식 대신, User -> Folder -> Topic -> Presentation으로 이어지는 계층형 컬렉션 구조를 채택하여 데이터를 직관적으로 관리하고, 읽기/쓰기 성능을 최적화하였다.

2. 개발 환경 이슈 및 서버-클라이언트 분리 개발

- 문제점: Android Studio 에뮬레이터의 리소스 점유율 문제로 인해 로컬 서버(FastAPI)와 클라이언트 앱을 동시에 구동하여 테스트할 때 잦은 렉과 네트워크 연결 끊김 현상이 발생하였다. 이는 개발 생산성을 저하시키는 주요 원인이 되었다.
- 해결 방안: 프론트엔드와 백엔드 개발 프로세스를 분리하는 전략을 취했다. 우선 FastAPI 서버를 독립적으로 구축하여 API 테스트 도구를 통해 데이터 처리 로직의 검증을 완료하였다. 이후 안정화된 서버 API를 Android 앱의 Retrofit 클라이언트와 연결하는 단계적 통합 방식을 적용하여 개발 효율성과 시스템 안정성을 확보하였다.

III. 결론

3.1 프로젝트 결과 요약

본 팀은 계획서에서 제안한 바와 같이, 영상과 음성을 복합적으로 분석하는 멀티모달 발표 평가 시스템을 성공적으로 개발하였다.

- 기능적 완성도:** 발표 영상 업로드부터 분석, 점수 산출, 리포트 생성까지의 전 과정이 끊김 없이 동작함을 확인하였다.
- 기술적 성취:** 최신 AI 기술(Whisper, MediaPipe)을 하나의 파이프라인으로 통합하여 실제 서비스 가능한 수준의 시스템을 구축하였다.

3.2 목표 달성을 평가

목표	중요도(%)	달성도(%)	수행내용
멀티모달 융합 분석	30	95	시선, 표정, 음성, 텍스트 데이터를 통합하는 분석 파이프라인을 구축하였고, FFmpeg–MediaPipe–Whisper–Praat–GPT를 연계하여 정량적 점수 산출까지 엔드투엔드로 동작함을 확인하였다.
사용자 앱(UI/UX) 구현	30	80	로그인, 폴더 관리, 영상 업로드, 분석 진행 상태 확인, 결과 화면 등 핵심 사용자 흐름은 모두 구현되어 실제 사용이 가능하다. 다만, 마이페이지에서 개별 분석 결과 화면으로 바로 진입하는 기능과 Script 대본 챗봇의 서버 연동은 미구현 상태로 남아 있어 향후 보완이 필요하다.
결과 시각화 및 리포트	20	85	도넛 차트를 활용한 시각화와 PDF/Excel 자동 리포트 생성 기능은 정상 동작하며, 동일 주제 내 여러 팀의 점수를 엑셀 파일로 비교할 수 있다. 그러나 분석 결과 화면에서 원본 발표 영상을 바로 재생하는 기능은 플레이어 UI만 구현된 상태로, 스트리밍/파일 경로 연동은 차기 과제로 남겼다.
AI 모델 정확도 확보	20	90	Whisper(STT)는 실제 테스트에서 한국어 인식률 90% 이상을 보였으며, MediaPipe를 기반으로 한 얼굴 랜드마크 추적도 안정적으로 동작한다.
합계	100	90	

3.3 향후 발전 방향

본 프로젝트는 멀티모달 AI 발표 평가 시스템의 핵심 기능 구현을 통해 서비스 가능 수준의 프로토타입을 구축하였다. 다만, 프로젝트 계획서에서 제시한 장기 목표와 비교했을 때 아직 완성되지 않은 기능들이 존재하며, 해당 기능들은 차기학기 캡스톤디자인 II에서 연속 개발할 예정이다.

1. AI 채점 엔진 고도화 및 딥러닝 자체 학습 모델 구축

- 현재는 OpenAI 기반 채점 엔진에 의존
- 계획서에서는 누적 데이터를 활용한 독자 모델 개발 및 학습 목표 제시
→ 실제 사용자 데이터를 활용해 발표 성능 향상 요소를 직접 학습하는 방향으로 확장
- GPT 의존도 최소화 → 운영비 절감 및 모델 신뢰성 확보

2. 프론트엔드 주요 기능 완성 및 사용자 경험 개선

기능	현 상태	향후 개발
마이페이지 분석기록 관리	UI 구현 및 Firebase 연동	Firebase와 연동하여 분석 결과 조회 적용
발표 결과 화면 영상 재생		서버 저장 영상 스트리밍 연동
Script(대본 챗봇)		추론 API 연결 및 실시간 대본 피드백 추가

IV. 참고 문헌

[1] OpenAI, "Whisper: Multilingual Speech Recognition Model," 2023.

<https://github.com/openai/whisper>

[2] Google Developers, "MediaPipe Face Landmarker Solution," 2024.

<https://developers.google.com/mediapipe>

[3] Bronkhorst, A. W., "Parselmouth: A Praat interface in Python," 2021.

<https://parselmouth.readthedocs.io/>

[4] FastAPI Documentation – High Performance Async Framework, 2024.

<https://fastapi.tiangolo.com/>

[5] Android Developers, "Kotlin and Jetpack Libraries," 2024.

<https://developer.android.com/kotlin>

[6] iText Library Official Documentation, 2024.

<https://itextpdf.com/en/resources/api-documentation>

[부 록]

소스코드 주요 구조

- **Android 앱 구조 (Kotlin/MVVM)**

```
com.minyook.overnight
|   └── ui
|       ├── activity
|       |   ├── OvernightActivity.kt          # 메인 네비게이션 컨트롤러
|       |   ├── PresentationInfoActivity.kt    # 발표 정보 입력
|       |   ├── UploadActivity.kt             # 영상 업로드/서버 요청
|       |   ├── AnalysisProgressActivity.kt    # 분석 상태 풀링
|       |   ├── AnalysisResultActivity.kt      # 결과 화면
|       |   ├── ScriptChatActivity.kt          # 발표 대본 챗봇 UI
|       |   └── GuideActivity.kt                # 서비스 이용 가이드
|       ├── fragment
|       |   ├── HomeFragment.kt
|       |   ├── FolderFragment.kt
|       |   └── MyPageFragment.kt
|       ├── adapter
|       |   ├── CriteriaListAdapter.kt         # 채점 기준 RecyclerView
|       |   └── ChatAdapter.kt                # 챗봇 메시지 RecyclerView
|       ├── customview
|       |   └── MultiSegmentDonutChart.kt     # 커스텀 도넛 차트 뷰
|       |
|       └── data
|           ├── model
|           |   ├── User.kt
|           |   ├── PresentationFile.kt
|           |   ├── CriterionResult.kt
|           |   └── AnalysisResult.kt
|           ├── network
|           |   ├── RetrofitClient.kt           # 백엔드 서버 통신
|           |   └── ApiService.kt              # /analyze, /status, /chat API 인터페이스
|           |
|           └── utils
|               └── FileUtils.kt / PdfUtils.kt # 로컬 파일 관리, PDF 저장
```

- **FastAPI 서버 구조 (Python / AI 분석 파이프라인)**

```
backend/
  └── main.py                      # FastAPI 엔트리포인트(CORS, 라우터 설정)
  └── processing/
    ├── task_manager.py            # 비동기 파이프라인 관리, job_id 상태 관리
    ├── video_analyzer.py         # FFmpeg로 프레임 추출
    ├── face_analyzer.py          # MediaPipe 얼굴 분석(표정/시선)
    ├── audio_analyzer.py         # Whisper STT + Prosody 분석
    ├── data_combiner.py          # 시각/청각 융합 + 세그먼트 정렬
    └── ai_scorer.py              # GPT-4o-mini로 채점 JSON 생성
  └── utils/
    ├── helpers.py                # 파일 저장, 디렉토리/임시파일 클린업
    └── json_helpers.py           # 채점 기준 JSON 저장/로드
  └── static/
    └── index.html / chat.html    # 웹 테스트 UI
└── .env / requirements.txt
```

[소감]

임진호 (20212979)	<p>이번 캡스톤디자인 I에서 서버 백엔드와 멀티모달 AI 분석 파이프라인 구축을 담당했다. 발표 영상을 서버에서 받아 FFmpeg로 프레임과 오디오를 분리하고, MediaPipe를 이용해 시선·표정·얼굴 움직임을 수치화 했으며, Whisper를 기반으로 음성을 텍스트로 변환하고 지터·거칠기 같은 음성 특성까지 분석하는 전체 파이프라인을 직접 구현했습니다. 특히 실제 발표자의 발음 오류나 말버릇 때문에 텍스트가 부정확하게 변환되는 문제를 해결하기 위해 전처리 기능을 도입했다가, 곧 중요한 사실을 깨달았다. 발음을 보정해 버리면 발음 자체에 대한 평가 기준을 잃어버리는 역효과가 생긴다는 점이었다. 결국 전처리 로직을 과감히 제거하고, 모델이 수행해야 하는 영역과 사람이 판단해야 하는 영역을 다시 정의하는 경험을 통해 설계 관점이 한층 성숙해졌다.</p> <p>또한 Whisper–MediaPipe–Praat에서 추출한 다중 데이터를 타임스탬프로 정렬해 GPT로 넘기는 과정에서, 어떤 데이터를 어떤 방식으로 전달 해야 가장 객관적인 평가가 가능한지 수없이 테스트하며 프롬프트 구조를 다듬었다. 이 과정은 단순한 개발을 넘어, 시스템이 가진 한계와 가능성을 동시에 체감할 수 있었던 값진 시간이었다.</p> <p>현재 채점 엔진은 GPT 기반이지만, 장기적으로는 누적 발표 데이터를 활용해 우리 팀만의 모델을 직접 학습시키는 것이 목표다. STT·표정 분석·음성 특징을 모두 포함한 진정한 “자체 채점 AI”를 만드는 것은 쉽지 않겠지만, 이번 프로젝트를 통해 그 기반을 확실히 다졌다고 생각한다.</p> <p>다음 학기 캡스톤디자인 II에서는 지금 구축한 파이프라인을 더욱 안정화하고, GPT 의존도를 낮춘 독자적인 AI 평가 모델을 개발해 완성도 높은 서비스로 발전시키는 것이 목표이다.</p>
임민욱 (20213034)	<p>이번에 캡스톤 디자인에서 프론트엔드의 어플리케이션 UI를 구현하였습니다. 평소에 코틀린이라는 언어를 가지고 Android 어플리케이션을 개인 프로젝트를 진행하였습니다. 하지만 이번엔 캡스톤 디자인을 통해서 마음이 잘맞는 팀원들을 만나서 프론트와 백엔드의 구조를 확실하게 갖추어서 진행을 하였습니다. 오랜만에 이 구조를 가지고 개발을 하였고 또한 비전 AI를 가지고 개발하는 것은 처음이라 설레는 마음을 가지고 개발하였습니다. 초반엔 프론트엔드에서는 디자인에 대해서 많은 의견이 왔다갔다하여 디자인과 화면 파이프라인을 구축하는게 가장 어려웠던 점이었습니다. 팀원들과 회의를 하면서 이런 점이 좋고 저런 점은 안좋았다는 피드백이 오가면서 2주일이라는 시간이 걸려서 저희만의 UI 화면 디자인을 완성했습니다. 팀원들도 만족을 하였고 뿐만 아니라 현대식에 맞춰 전체적으로 깔끔하고 모던한 느낌의 부드러운 면이 추가가 되어서 좋은 화면이 나왔습니다. 이후에 Google에서 만든 Firebase라는 데이터베이스, Auth, Cloud, Storage, AI 등등의 많은 것들을 지원을 해주는 플랫폼을 사용하여 Auth라는 서비스로 로그인/회원가입을 Firebase Firestore라는 NoSQL 구조를 가진 데이터 베이스를 사용하여 저희의 어플리케이션을 연동하였습니다. 이러한 것들은 제가 이전 개인 프로젝트를 통해서 얻은 프론트만으로 어플리케이션을 구축하는 방법인데 이 방법을 통해서 팀원들에게 알</p>

	<p>려주고 알려준 방식대로 개발을 하여 빠른 시간 내에 개발을 성공적으로 하게 되어서 큰 문제없이 끝나서 기분이 좋았고 한편으로는 뿌듯한 마음도 있었습니다. 캡스톤 디자인으로 혼자가 아닌 팀원들과 좀 더 체계적인 개발을 하게 되어서 좋은 경험을 하였고 새로운 목표가 생기는 시작점이라고 생각합니다. 이후로도 저희가 정해놓은 목표를 멈추지 않고 도달하는 마음가짐으로 프로젝트에 임할 수 있는 계기가 된 것 같습니다.</p>
<p>이규찬 (20212977)</p>	<p>지난 여름방학, 음성 텍스트 변환(STT) 기술만을 활용해 발표 내용을 채점하는 웹 서비스를 개발했었습니다. 하지만 '발표'라는 것은 단순히 대부분의 내용뿐만 아니라, 발표자의 시선 처리, 목소리의 떨림, 태도 등 비언어적 요소가 결합된 종합적인 커뮤니케이션이라는 점을 깨달았습니다. 이번 캡스톤디자인 I에서는 이러한 한계를 극복하고자 '비전 기반의 멀티모달 분석'을 도입했습니다. 특히 익숙한 웹 환경을 벗어나 안드로이드 스튜디오와 코틀린을 처음 학습하며 개발해야 했기에 기술적인 부분에서 다소 어려움이 있었지만, 이를 극복하고 시선 추적과 음성 떨림 분석 기능을 모바일 환경에 성공적으로 구현해냈습니다. 단순 텍스트 분석을 넘어, 발표자의 행동 패턴까지 입체적으로 피드백해주는 '진짜 발표 코치'를 완성하며 개발자로서 한 단계 도약할 수 있었던 뜻깊은 프로젝트였습니다.</p>
<p>이지민 (20233065)</p>	<p>이번 캡스톤디자인 I은 웹 개발에 익숙했던 저에게 모바일 앱 개발이라는 새로운 분야에 도전할 수 있는 뜻깊은 계기가 되었습니다. 안드로이드 스튜디오와 코틀린을 처음 접하다 보니 레이아웃 구성 단계부터 많은 시행착오를 겪었지만, 발생하는 오류들을 하나씩 해결해 나가는 과정에서 안드로이드 시스템의 구조를 깊이 이해하고 개발자로서의 역량을 넓힐 수 있었습니다. 특히 프론트엔드를 전담하면서, 단순히 기능이 작동하는 화면을 구현하는 것을 넘어 철저히 '사용자 입장'에서 서비스를 바라보는 시각을 갖게 되었습니다. 사용자의 행동 흐름을 예측하고, 필요한 정보를 적재적소에 배치하기 위해 고민했던 시간들이 큰 배움이 되었습니다. 이번 학기의 경험을 발판 삼아, 다음 학기에는 부족한 부분들을 보완하여 Overnight.AI를 완성도 높은 서비스로 발전시키겠습니다.</p>