

# 제8회 산학연계 SW프로젝트 최종보고서

팀명	씹크빅데이터
프로젝트 수행기간	2024. 03. 01 ~ 2024. 10. 31
프로젝트 주제	딥러닝 데이터 분석을 이용한 추천 시스템 개발 (CleVo : AI를 활용한 영어 발음 학습 서비스 개발)
지도 교수	정보융합학부 임동혁 교수
참여업체 명	(주) 핀다

2024. 11. 04.



**광운대학교**  
KwangWoon University

# 산학연계SW프로젝트 최종보고서

팀 명	딥러닝 데이터 분석을 이용한 추천 시스템 개발			
과제 명	씹크빅데이터			
GitHub URL	<a href="https://github.com/warmcotton/ThinkBigData_CleVo">https://github.com/warmcotton/ThinkBigData_CleVo</a>			
YouTube URL	<a href="https://youtu.be/yWXYF2xx75g">https://youtu.be/yWXYF2xx75g</a>			
수행기간	2024년 03월 01일 ~ 2024년 10월 31일			
과제비	총 1,200,000 원			
지도교수	성 명	임 동 혁	학 부	정보융합학부
참여학생	성 명	학 부	학 번	email
	오승준	정보융합학부	2019204014	joon7513@naver.com
	이원빈	정보융합학부	2019204065	johnlee08230823@gmail.com
	신원석	정보융합학부	2018204057	wonseok0807@naver.com
	김민서	정보융합학부	2020204047	dia777mond@naver.com
	설수빈	정보융합학부	2020204050	tjftnqls124@gmail.com
참여업체	회사명	(주) 핀다	담당자	전 희 국
	연락처	010-9988-0922	email	heegook@finda.co.kr
『산학연계SW프로젝트』지원계획에 따라 최종보고서를 제출합니다.				
2024년 11월 4일				
	팀 장	오 승 준	모 (원주)	
	팀 원	이 원 빈	이원빈 (인)	
	팀 원	신 원 석	신원석 (인)	
	팀 원	김 민 서	김민서 (인)	
	팀 원	설 수 빈	설수빈 (인)	
	지도교수	임 동 혁	임동혁 (인)	
광운대학교 소프트웨어융합대학 귀하				

# 목 차

1. 과제의 개요 .....	
가. 배경 및 필요성 .....	4
나. 목표 .....	4
다. 개발 내용 .....	5
2. 과제의 내용 .....	
가. 설계 및 개발의 내용 .....	6
나. 수행 방법 및 추진 과정 .....	15
다. 최종 결과물 .....	17
라. 소프트웨어 등록 .....	18
마. 예산 집행 .....	18
바. 개선 방안 .....	18
3. 오픈소스SW 활용 및 기여 .....	
가. 오픈소스SW 활용 .....	19
나. 오픈소스SW 기여 .....	20
4. 과제의 향후 계획 .....	
가. 활용 방안 .....	21
나. 기대 효과 .....	21
5. 별첨 .....	
가. 저작권 등록증 .....	22
나. 전시회 패널 자료 .....	23
다. 유튜브 스냅샷 .....	24
라. 서비스 실행 화면 .....	25

## 1. 과제의 개요

### 가. 배경 및 필요성

프로젝트의 주제가 '딥러닝 데이터 분석을 활용한 추천 시스템 개발'인 만큼, 프로젝트의 주제를 선정함에 있어 가장 큰 고려 사항은 가능한 많은 사용자들이 공감할 수 있는 Pain Point로부터 출발하는 것이었다. 따라서, 이를 만족시킬 수 있는 주제에 대해 다양한 고민을 해 본 결과, 대다수의 한국인 영어 학습자들이 Reading 또는 Listening과 관련된 부분에서는 강세를 보이는 반면, Speaking과 관련해서는 학습에 대체로 많은 어려움을 겪고 있다는 점을 확인할 수 있었다. (김희경(2010), 영어발음교육에 대한 한국인 영어학습자의 인식) 그리고, 이러한 현상에 대한 근본적인 원인은 Reading과 Listening에 치우친 교육과정 이외에도 Speaking 영역의 특성상 학습자에게 적절한 학습 자료와 잦은 피드백이 제공되기 어렵다는 데에서 찾을 수 있었는데, 바로 이러한 부분에서 사용자에게 적절한 학습 자료를 추천해주는 딥러닝 모델의 필요성을 떠올릴 수 있었다.

따라서, 본 프로젝트에서는 'AI를 활용해서 한국인 영어 학습자의 Speaking 학습을 보조하자'라는 목표를 세우게 되었고. 그 결과, 다양한 Speaking 영역 중에서도 특히 가장 학습하기 어렵다고 판단되는 '발음'이라는 영역에 집중하여, 관련된 논문(구자현(2018), 한국인 영어 학습자를 위한 효과적인 발음 지도 연구)에서 설명하는 효과적인 지도 방식인 '피드백 제공'과 '유사한 발음 반복'이라는 기능을 제공하는 영어 학습 보조 서비스, 'AI를 활용한 영어 발음 학습 서비스 CleVo'를 기획하게 되었다.

### 나. 목표

본 프로젝트는 '사용자가 소리내어 읽은 문장을 AI가 분석하여 Scoring 한 뒤, 취약한 부분을 개선하는데 도움이 될 수 있는 문장을 추천해주는 Web 기반 영어 학습 서비스'의 구현을 목표로 진행되며, 이를 위해 설정된 서비스 목표와 이를 이루기 위한 개발 목표는 다음과 같다.

#### • 서비스 목표

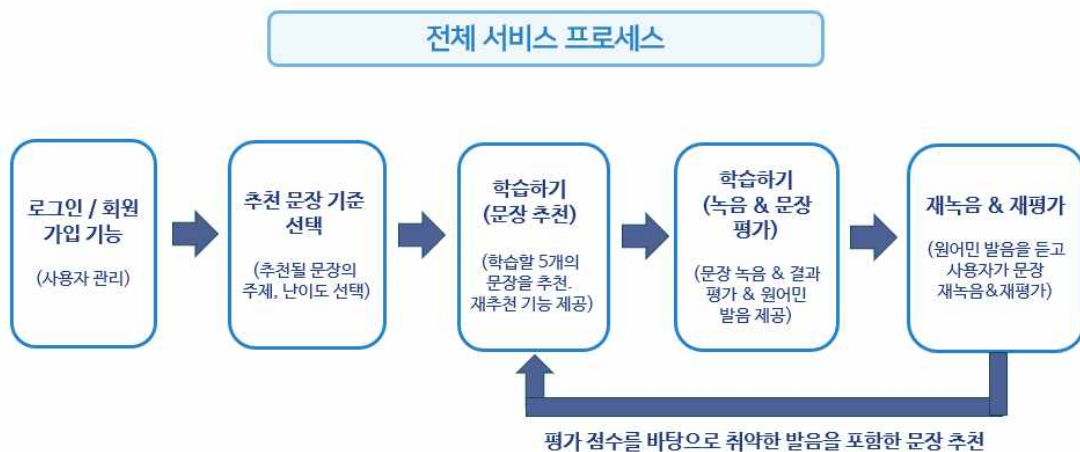
- 적절한 발음 평가 : 발음 평가 모델은 사용자의 발음을 분석하여, 적절한 Scoring과 취약점을 발견할 수 있어야 한다.
- 문장 추천의 정확성 : 문장 추천 모델은 사용자의 발음 정보와 취약 발음 정보, 사용자의 관심사, 선택 난이도 등의 정보를 바탕으로, 사용자의 발음 학습에 실질적으로 도움이 될 수 있는 문장들을 추천할 수 있어야 한다.
- 학습 보조 기능 : 사용자가 Self-Feedback을 통해 발음 개선을 주도할 수 있도록, 회원 관리 기능(로그인, 회원가입, 아이디/비밀번호 찾기 등), 원어민 발음 제공, 문장 보관, 단어 검색, 이전 학습 로그, 학습 진도율 관리 등의 학습 관련 보조 기능들이 제공되어야 한다.
- 접근성 : 사용자가 서비스 접근에 어려움이 없도록 간단하고 직관적인 페이지 디자인 & 기능 설계가 이루어져야 한다.

## • 개발 목표

- 적절한 모델 API의 활용 : 적절한 발음 평가 모델의 성능을 위해 적절한 성능과 배포의 안정성이 보장된 ETRI의 발음 평가 API 모델을 활용한다.
- 추천 모델 고도화 : 서비스의 주요 기능을 구현하기 위해 발음 평가 모델로부터 발음 평가 결과 이외에도 취약 발음을 추출한 다음, 이를 바탕으로 적절한 문장 추천이 이루어질 수 있도록 LLM과 Prompt Engineering을 활용하여 다양한 비교 실험을 진행한 뒤, 결과를 바탕으로 최적의 모델 + 프롬프트 조합을 서비스에 채택한다.
- 추천 성능 평가 방식 : 추천 모델의 성능 평가에 있어, 적절한 평가가 이루어질 수 있도록 다양한 측면에서의 추천 성능을 평가하고 이를 비교한다.
- 웹사이트 제작 : 하나의 서비스로써 기능할 수 있도록, 사용자의 편의성을 고려한 웹사이트를 Spring Boot 기반 서버와 JavaScript/CSS/Html 기반의 웹서비스로 구현한다.
- 다양한 기능 구현 : 웹서비스에서 제공하는 학습 외 부가 기능들에 대해서도 모든 기능들이 안정적으로 작동할 수 있도록 다양한 예외 상황들에 대응할 수 있는 대책을 마련해 둔다.

## 다. 개발의 내용

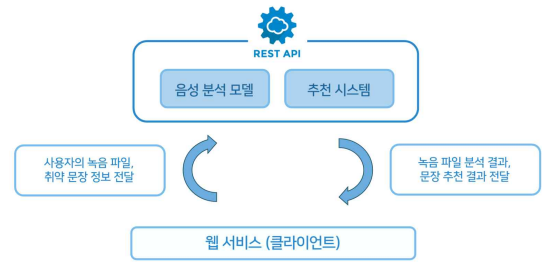
본 서비스의 핵심 기능은 아래와 같이 요약될 수 있으며, 이를 위한 서비스 프로세스는 [그림 1]과 같다.



[그림 1]

이때의 '학습하기' 과정에서는 '발음 평가 모델'과 '문장 추천 모델'이라는 2개의 딥러닝 모델이 사용되며, 웹 서비스를 통한 사용자의 요청(클라이언트)에 따른 모델의 출력 결과를 원활하게 전달하기 위해서 RESTful API 형태로 모델을 배포하게 된다. 이를 도식화한 결과는 [그림 2]와 같다.

이러한 주요 기능들은 웹사이트 형식으로 사용자들에게 배포 및 제공되며, 이를 구현하는 과정에서 서버는 SpringBoot, DB는 MySQL, 웹페이지는 HTML/CSS/JavaScript, 추천 시스템 모델은 ChatGPT 3.5 Turbo, 배포는 AWS Lambda, 음성 평가 모델은 ETRI에서 제공하는 '한국인의 영어 발음 평가 모델 API'가 사용되었다.



[그림 2]

## 2. 과제의 내용

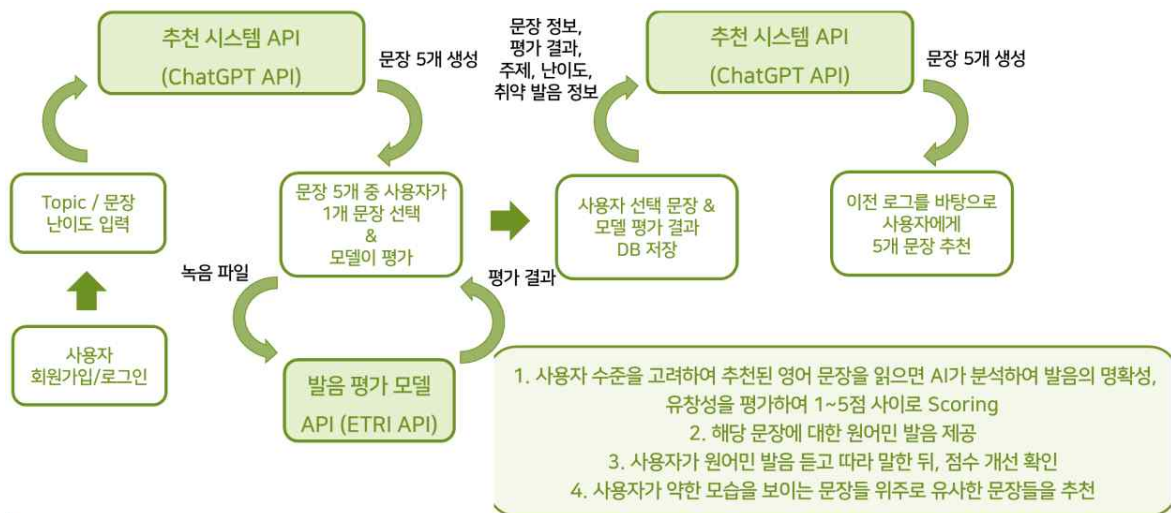
### 가. 설계 및 개발의 내용

#### 1) 개념 설계 (구조 설계)

초기에 기획했던 '영어 발음 학습에 도움이 될 수 있는 문장을 추천할 수 있는 웹서비스' 구현을 위해서는 '발음 평가 - 문장 추천'이라는 '학습하기' 메커니즘 이외에도 회원 관리, 문장 보관, 단어 검색, 발음 제공, 메모장 등의 '학습 보조 기능'들이 제공될 필요가 있었다. 이를 위해 최종적으로 구현된 '학습 보조 기능'들은 [표 1]과 같으며, '학습하기 기능'의 메커니즘은 [그림 3]과 같이 구현되었다.

학습 보조 기능	
<ul style="list-style-type: none"> <li>• 회원가입 / 로그인</li> <li>• 비밀번호 찾기</li> <li>• 회원 정보 수정</li> <li>• 관심 주제 / 문장 난이도 수집</li> <li>• 데일리 학습 진도율 관리, 메모장</li> </ul>	<ul style="list-style-type: none"> <li>• 재학습 / 재녹음 기능</li> <li>• 단어 검색 기능</li> <li>• 원어민 발음 듣기</li> <li>• 녹음 파일 다운로드</li> <li>• 문장 보관 (학습 로그 관리)</li> </ul>

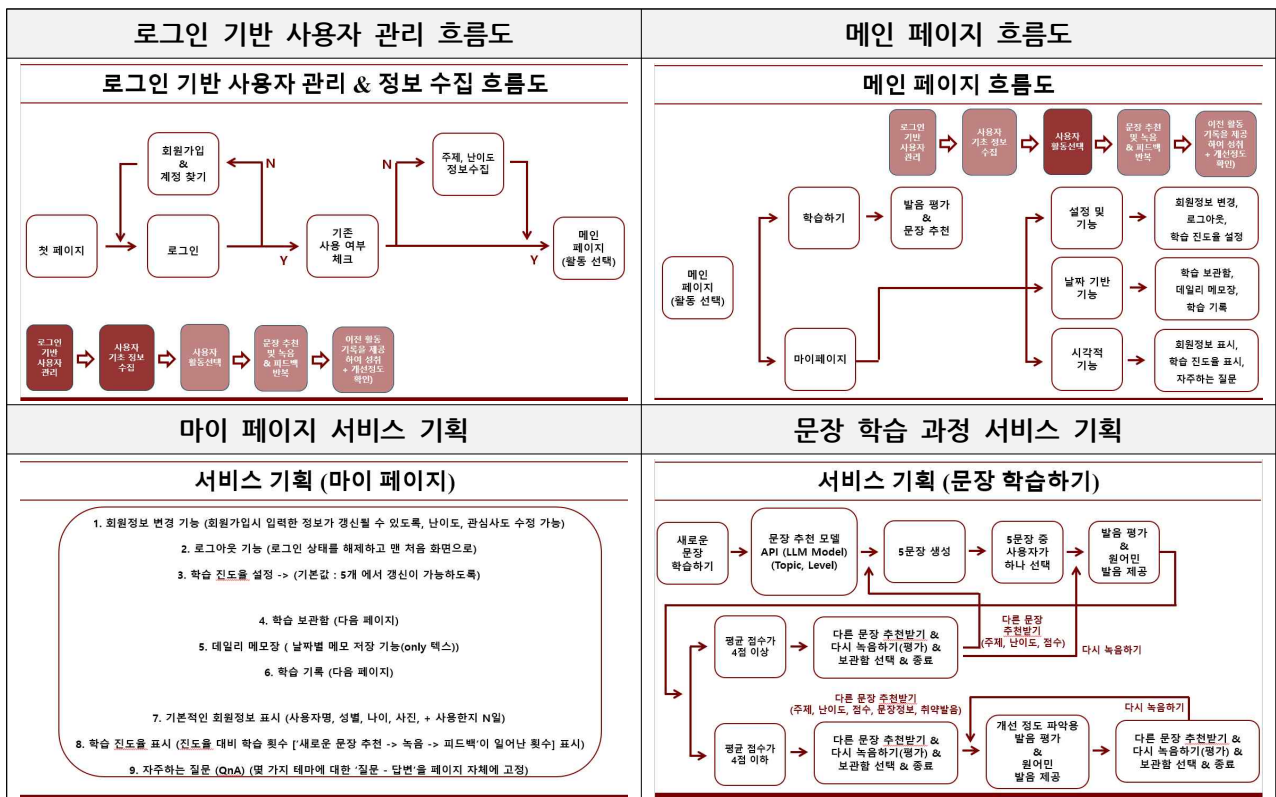
[표 1]



[그림 3]

이는 기본적인 로그인 / 회원가입 기능을 통해 회원 정보를 DB에 저장 후, 추천될 문장에 사용자의 학습 목적이 반영될 수 있도록 사용자가 선택한 '주제'와 '문장 길이(난이도)' 정보를 입력받아 이 역시 회원 DB에 저장 후, 앞선 정보들을 바탕으로 문장 추천 모델이 5개의 문장을 생성하는 것을 시작으로 학습하기 기능의 문장 추천에서 발생하는 Cold-Start 문제를 해결하였고, 이후의 문장 추천에 있어서는 앞선 정보 이외에도 사용자의 발음 평가 정보(유창성, 정확성)와 문장 정보, 취약 발음 정보를 반영하여 문장 추천을 진행하게 된다.

로그인 과정부터 시작하는 전체 서비스 흐름도는 [그림 4]와 같다.



[그림 4]

추가적으로, 최종적으로 서비스를 구현하는 과정은 [그림 5]와 같이 AI/Back-End/Front-End 파트가 각자의 역할을 나눈 뒤, 각각의 역할에 따라 만들어진 서버, 웹페이지, 모델 API 결과물을 합치는 방식으로 진행하였다.



[그림 5]

## 2) 상세 설계 (기능 설계)

### ① 서비스 디자인 기획 & 페이지 제작

발음 정보를 AI가 분석하고, 취약점을 개선할 수 있는 솔루션을 제공한다는 점에서 CleverVoice라는 키워드를 줄인 'CleVo'를 서비스명으로 사용하였다.

이때, 로고의 점점 진해지는 색은 서비스를 통해 점점 성장하는 사용자를 표현하고자 하였으며, 페이지 디자인 역시 로고와 잘 어울릴 수 있도록 기획하였다.



CleVo 서비스 로고



CleVo 서비스 페이지

### ② 회원 관리 기능

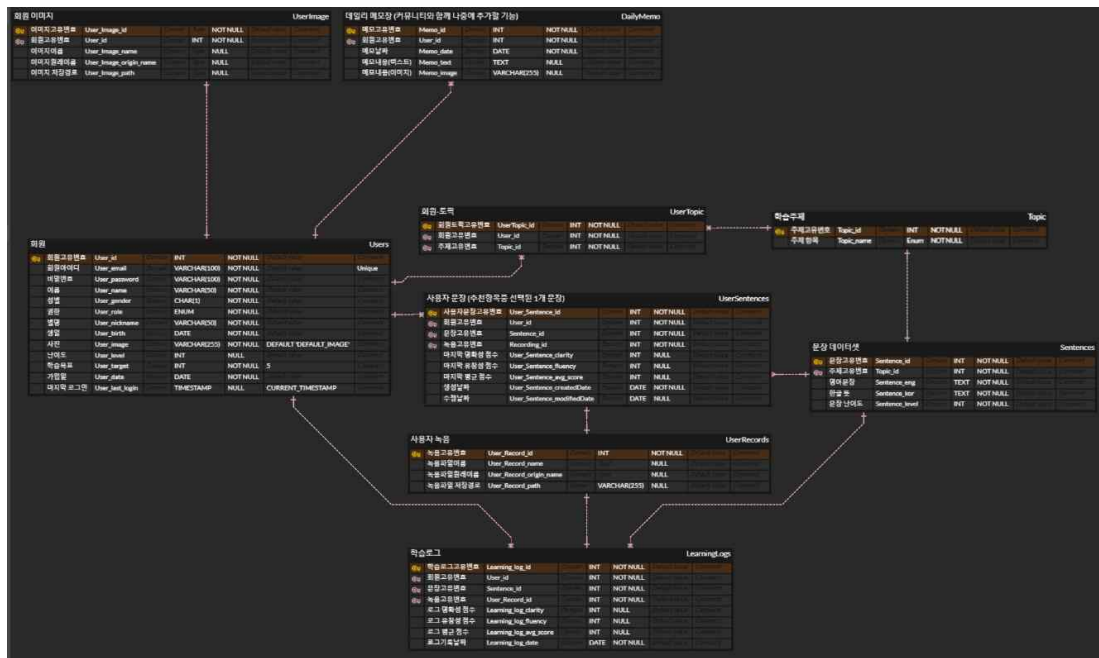
SpringBoot 환경에서 백엔드 기능을 개발했으며, 사용자 맞춤형 문장 추천을 하기 위해 회원 관리 기능을 개발하였다. 해당 기능을 통해 각 회원별로 추천된 문장이 DB에 저장되며, 문장 추천 모델과의 통신에 사용할 DTO(Data Transfer Object)를 설계 및 구현하였다.

이때, 회원가입 과정에서 사용자로부터 수집하여 DB에서 관리하는 정보들은 다음과 같다.

사용자로부터 수집한 정보			
• 이메일	• 별명	• 생년월일	• 문장 주제
• 비밀번호	• 이름	• 성별	• 문장 난이도

사용자 관련 데이터

구현된 DB



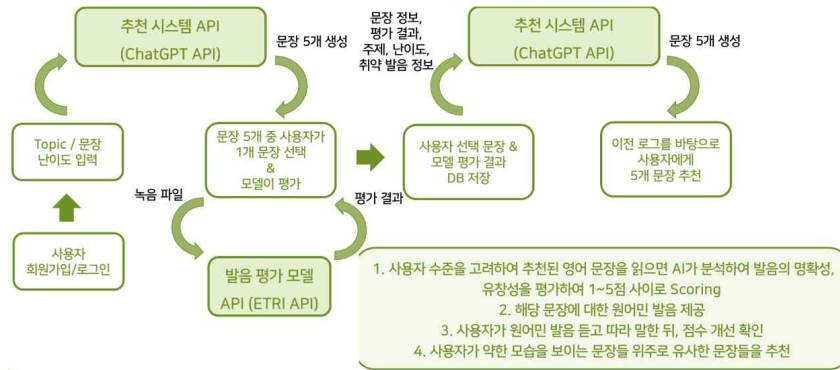
전체 서비스 ERD

<https://www.ercloud.com/d/TSz4jXndWedm56Eu6>



### ③ 학습하기 기능

본 프로토타입에서의 '학습하기' 기능은 [그림 6]와 같은 순서로 진행된다. 이는 ChatGPT API 기반의 추천 시스템 API와 ETRI에서 제공하는 발음 평가 모델 API를 호출하는 방식으로 구현되었다.



[그림 6]

학습하기의 전체적인 프로세스는 사용자가 선택한 주제/난이도 정보를 바탕으로 LLM 모델(추천 시스템 모델)이 생성한 5개의 문장(Cold-Start) 중, 사용자가 선택한 학습 문장에 대한 발음이 평가 모델에 전달되어 평가 결과를 얻게 되면, 다음 LLM 모델이 문장을 생성할 때는 이전의 평가 결과(유창성 점수, 정확성 점수, 문장 정보, 취약 발음 정보, 주제, 난이도)를 고려하도록 생성하는 과정으로 이루어진다.

이외에도 학습의 보조를 위해 Google TTS(Text To Speech) API를 활용해 해당 문장에 대한 원어민 발음을 제공하는 기능과 사용자의 녹음된 발음을 다운로드 할 수 있는 기능 또한 구현되었다.

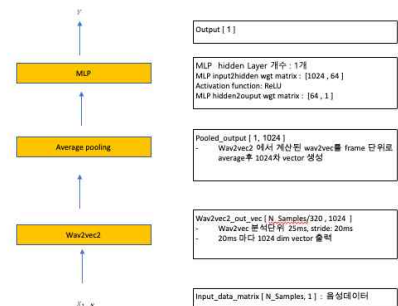


학습하기 화면

### ④ 발음 평가 모델 (유창성 점수)

본 프로젝트에서 사용한 발음 평가 모델은 한국전자통신연구원(이하 ETRI)에서 공개한 발음평가 API를 활용하였다. 이는 AI HUB에서 제공하는 한국외국어대학교&엠브레인퍼블릭의 전문가들이 한국인의 영어 발음에 대해 1~5점 사이로 Labeling한 데이터에 대해 0.7 만큼의 Correlation Score라는 성능을 보여주었으며, RESTful API 형태로 공개되었기에 자체 서버 내에서 이를 호출할 수 있었다.

서비스에서는 사용자에게 음성을 녹음받고, 적절한 발음 평가를 제공해주기 위해 JavaScript로 음성 파일을 얻은 뒤, base64형식으로 인코딩하여 백엔드와 통신하도록 하였으며, 서버에서는 해당 데이터를 평가 모델과의 통신을 통해 평가하고 이를 학습용 DTO에 저장 후 프론트엔드에게 넘겨준 뒤, 필요한 정보들을 JPA 기반 DataBase에 저장하는 방식으로 이를



발음 평가 모델 구조

구현하였다. 모델 구조 (음성 데이터를 벡터화하기 위한 Wav2Vec 모델을 기반으로 'Wav2Vec -> Pooling -> Task를 위한 MLP' 구조) 이렇게 구해진 평가 결과는 본 프로젝트에서 사용자의 발음에 대한 '유창성 점수'로써 기능하게 된다.

### ⑤ 발음 평가 모델 (정확성 점수와 취약 발음)

ETRI에서 제공하는 발음 평가 API는 사용자의 발음에 대한 평가 결과를 제공할 뿐만 아니라, 해당 발음을 AI가 인식한 결과를 문자열로 반환해주는 '인식 결과' 또한 제공한다. 사용자의 발음으로부터 추출한 '정확성 점수'와 '취약 발음'은 이 '인식 결과'를 통해서 구하게 되는데 그 과정은 다음과 같다.



발음 평가 API의 Input/Output

정확성 점수 계산 과정
<ol style="list-style-type: none"> <li>1. 사용자가 선택한 '문장 정보'와 발음 평가 API로부터 전달받은 '인식 결과'를 저장한다.</li> <li>2. 두 문자열을 모두 CMU Pronouncing Dictionary (cmudict-0.7b)를 활용하여 CMU 음소로 변환한다.</li> <li>3. 변환한 두 음소 문자열에 대해 레벤슈타인 거리(Levenshtein Distance)를 계산한다.</li> <li>4. 앞선 레벤슈타인 거리 정보를 바탕으로 PER(Phoneme Error Rate)을 계산한다.</li> <li>5. 해당 PER을 바탕으로 다음 기준에 따라 1.0~5.0 사이의 점수를 계산하고, 이 점수를 '정확성 점수'로 정한다. <ul style="list-style-type: none"> <li>- 3% 이하는 만점 5.0점</li> <li>- 10% 이하는 최하 4.5점, 15% 이하는 최하 4.0점</li> <li>- 25% 이하는 최하 3.5점, 40% 이하는 최하 3.0점</li> <li>- 60% 이하는 최하 2.5점, 70% 이하는 최하 2.0점</li> <li>- 80% 이하는 최하 1.5점, 90% 이하는 최하 1.0점</li> </ul> </li> </ol>
취약 발음 추출 과정
<ol style="list-style-type: none"> <li>1. 사용자가 선택한 '문장 정보'와 발음 평가 API로부터 전달받은 '인식 결과'를 저장한다.</li> <li>2. 두 문자열을 모두 CMU Pronouncing Dictionary (cmudict-0.7b)를 활용하여 CMU 음소로 변환한다.</li> <li>3. 두 음소 문자열을 비교하여 '취약 발음'을 추출한다. ('문장 정보'에 있지만, '인식 결과'에 없는 발음은 취약 발음)</li> <li>4. 취약 발음의 선정 기준은 다음과 같다. <ul style="list-style-type: none"> <li>- 오류 음소가 문장 정보에 존재하면 해당 음소는 우선으로 오류 음소 문자열에 포함한다.</li> <li>- 만약 원래 문장에는 없고, 발음한 문장에만 존재하는 음소이지만 그 빈도가 3회 이상이라면 이 또한 오류 음소 문자열에 포함한다.</li> <li>- 우선 순위에 있어서는 원래 문장에 존재하는 음소가 발음한 문장에만 존재하는 음소보다 우선순위가 높다.</li> </ul> </li> </ol>

Phoneme	Example	Phoneme	Example
aa	odd	k	key
ae	at	l	lee
ah	hut	m	me
ao	ought	n	knee
aw	cow	ng	ping
ay	hide	ow	oat

CMU 발음 예시



레벤슈타인 거리

이렇게 구해진 '문장 정보', '유창성 점수', '정확도 점수', '취약 발음' 정보는 이후 '문장 추천 (생성) 모델 API'의 파라미터로 전달되어, 생성 모델의 프롬프트에 직접적으로 반영된다.

## ⑥ 문장 추천 모델

서비스에서의 문장 추천은 LLM을 활용한 문장 추천 방식을 활용하였다. 이때의 LLM 모델은 서비스 자체에서 API를 제공하는 ChatGPT가 채택되었으며, 이후 서비스에서 Prompt Engineering을 통해 ChatGPT(3.5 Turbo) 모델이 원하는 기능을 수행하도록 만들기 위해 AWS Lambda 환경에서 ChatGPT API 라이브러리를 기반으로 프롬프트를 작성하고 응답을 받을 수 있는 코드를 작성하였고, 이를 API Gateway를 활용하여 RESTful API 형태로 배포하였다. 이때의 프롬프트는 아래와 같으며, 프롬프트에는 POST 방식으로 파라미터를 통해 전달받은 정보들(주제, 난이도(문장 길이), 문장 정보, 유창성 점수, 정확성 점수, 취약 발음)이 반영되어 사용자의 학습에 도움이 되는 문장들이 생성되게 된다.

```

"sentence": {
  "sen1": "Visiting museums in Europe is enriching.",
  "sen2": "How do you spend your weekends?",
  "sen3": "Reading books helps expand knowledge.",
  "sen4": "Cooking dinner with friends creates fun moments.",
  "sen5": "I watch TV shows before sleep."
},
"translations": {
  "sen_trans_1": "유럽의 박물관을 방문하는 것은 높은 가치가 있어요.",
  "sen_trans_2": "주말에 어떻게 보내나요?",
  "sen_trans_3": "책 읽기는 지식을 넓히는 데 도움이 돼요.",
  "sen_trans_4": "친구들과 함께 저녁 만들기는 즐거운 순간을 만들어요.",
  "sen_trans_5": "잠들기 전에 TV 프로그램을 봐요."
}

```

생성된 문장 정보 (API Return)

문장 추천 모델 API에 파라미터로 전달되는 정보	
첫 추천(Cold Start), 평가 점수가 4.0 이상인 경우	이전 평가 정보를 반영한 추천이 필요한 경우
주제(Topic), 문장 길이(length)	주제(Topic), 문장 길이(length), 문장 정보(reference), 유창성 점수(fluency_score), 정확성 점수(accuracy_score), 취약 발음 패턴(vulnerable_pronunciation)

(Topic : 취미생활, 비즈니스, 해외여행, 일상생활, 쇼핑)

(Length : 상 (10~15개 단어), 중 (7~10개 단어), 하 (7개 이하 단어))

최종적으로 프로젝트에 사용된 프롬프트는 다음과 같으며, 여기에는 다음과 같은 Prompt Engineering 기법들이 적용되었다.

- Role Setting
- Conditional Logic Response Formatting
- Meta-Learning

ChatGPT Prompt
<p><b>[역할 설정 &amp; 상황 제시]</b></p> <ul style="list-style-type: none"> <li>• {"role": "system", "content": "You are a helpful assistant who recommends 5 English sentences that can help improve the English pronunciation of Korean speakers. You should reflect the given topic and sentence length in your recommendations. Based on the received sentence and a score between 1 and 5, if the score is higher than 4, you will recommend a random sentence that reflects the given topic and length. If the score is lower than 4, you will recommend an appropriate sentence that has similar pronunciation or patterns to the given sentence."}</li> </ul> <p><b>[기능 설명 &amp; 출력 형태 지정]</b></p> <ul style="list-style-type: none"> <li>• {"role": "user", "content": "Generate 5 pairs of English sentences with their Korean translations. Each pair should be in the format: 'English sentence - Korean translation'. (Without Any Explanation &amp; Separator &amp; Index &amp; Order, Only Just Return 'value-value' pair)"}</li> </ul> <p><b>[출력 예시 제공]</b></p> <ul style="list-style-type: none"> <li>• {"role": "user", "content": "Response must not include details like ordering : '1. 2.', explaining: 'english-sentence', etc. I'll show you return example. 'Watching sunsets in Bali is mesmerizing.-발리의 일몰 감상 매혹적이에요.', 'Jogging in the park refreshes me.-공원에서 조깅하는 건 저를 기분 좋게 해줘요.', 'Pottery making is truly therapeutic.-도예 제작은 정말 치유적이에요.', 'Origami fascinates me endlessly.-접기에 끊임없는 매력을 느껴요.', 'Skiing in the Swiss Alps was thrilling.-스위스 알프스에서 스키를 탔어요.'"} </li> </ul> <p><b>[URL 파라미터로 전달받은 정보 고려 1(주제, 길이)]</b></p> <ul style="list-style-type: none"> <li>• {"role": "user", "content": "The theme of the recommended sentences should refer to the keyword '{topic}' and should not exceed '{length}' words. Additionally, the recommended sentences should include the CMU pronunciation element '{vulnerable_pronunciation}' to help users improve their pronunciation skills."}</li> </ul> <p><b>[URL 파라미터로 전달받은 정보 고려 2(점수, 문장), 예외처리]</b></p> <ul style="list-style-type: none"> <li>• {"role": "user", "content": "The previous user received a score of '{fluency_score}' for fluency and '{accuracy_score}' for accuracy. Based on these scores, you should generate sentences that appropriately refer to '{reference}' and include the pronunciation element '{vulnerable_pronunciation}' to help the user improve their learning. If '{reference}' or '{vulnerable_pronunciation}' is 'None, ignore them and recommend new and diverse sentences that satisfy the theme and word count conditions."}</li> </ul>

## ⑦ 문장 추천 모델의 평가

본 프로젝트의 목적을 잘 달성하기 위해서는 모델이 추천해준 문장의 적절성을 정량적으로 평가할 필요가 있었기에, 추천 모델의 성능 평가를 진행하였다.

성능 평가 기준은 모델에 정보를 전달했을 때 전달된 조건을 충분히 만족하고 있는지를 기준으로, 유창성 점수와 정확성 점수는 각각 1.0으로 설정하여 전달된 조건을 최대한으로 반영하고자 할 때 다음과 같은 목표를 가지고 설정되었다.

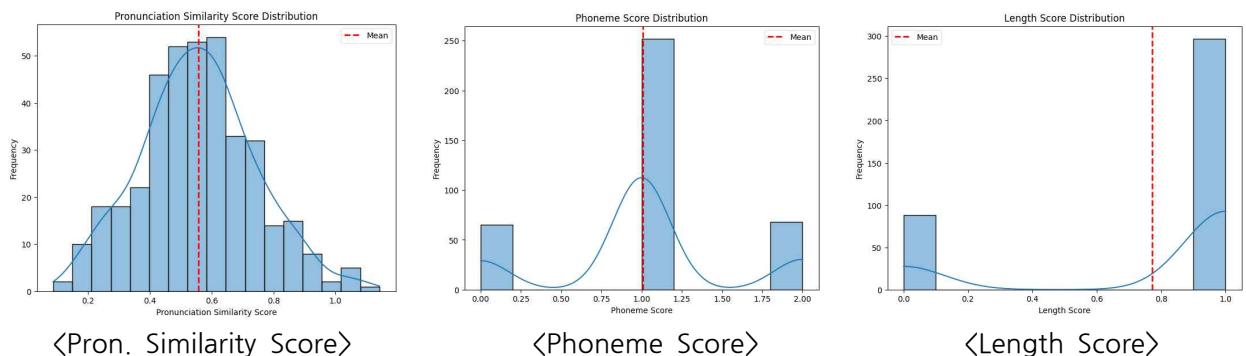
- 1) [Length Score] '제한된 문장 길이(난이도)를 만족하는 결과물을 생성해내는가'
- 2) [Pronunciation Similarity Score] '전달된 문장과 생성된 문장의 발음 유사도는 어느정도인가'
- 3) [Phoneme Score] '취약 발음은 충분히 반영된 결과물이 생성되었는가'

문장 추천(생성) 모델의 평가 방법은 다음과 같으며, ChatGPT 3.5 Turbo, ChatGPT 3.5 Turbo-16k, ChatGPT 4.0o 모델에 대한 각각의 성능지표는 그림과 같았다.

추천 모델 성능 평가 과정	
1. ChatGPT 4o를 활용하여 100개의 문장을 생성한다. (전체 주제 5개에 대해 각각 20개씩)	
2. 만들어진 문장에 대해 Length를 Mapping하고, 취약 발음에 대해서는 CMU 음소를 기준으로 랜덤하게 Labeling한다.	
3. 추천 모델 API에 만들어진 문장 정보와 Labeling 된 값들을 바탕으로 Reference, 문장 길이를 Length, 취약 발음을 Vulnerable_pronunciation, 유창성 점수와 정확성 점수를 모두 1.0으로 설정하여 전달하여 입력받은 100개의 문장마다 각각 5개의 문장을 생성하도록 한다. (총 500개의 생성 결과)	
4. Reference 문장과 생성된 문장을 바탕으로 다음과 같은 과정으로 성능지표를 계산한다.	
5. [Pronunciation Similarity Score] : 두 문장 중 길이가 긴 문장 대비, 일치하는 음소의 비율을 기준으로 0~2 사이 정규화 Mapping 한 값의 평균	
6. [Phoneme Score] : 생성 결과에 전달받은 취약 음소가 포함되지 않았다면 0, 1개 이상 포함되었다면 1, 모두 포함되었다면 2로 Mapping 한 값의 평균	
7. [Length Score] : 전달받은 문장 길이 조건에 부합하는 문장이 생성되었다면 1, 아니면 0을 출력한 결과의 평균	

ChatGPT 모델에 따른 성능 차이				
Model	Token	Pron. Similarity Score	Phoneme Score	Length Score
GPT 4.0o	128,000	0.53	0.97	<b>0.80</b>
GPT 3.5 Turbo-16k	16,385	<b>0.58</b>	0.98	0.58
<b>*GPT 3.5 Turbo*</b>	16,385	0.55	<b>1.00</b>	0.77

ChatGPT 3.5 Turbo 모델의 성능 평가 그래프

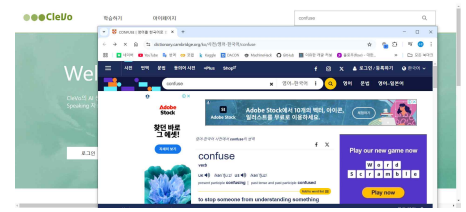


(Similarity Score는 정규 분포를 따르므로 대체로 발음이 적절히 유사한 문장들이 생성되었음을, Phoneme Score를 통해서는 대다수의 문장들은 최소 1개 이상 취약 발음을 포함하고 있음을, Length Score를 통해서는 약 77%의 확률로 문장 길이 조건을 만족하는 것을 알 수 있다.)

따라서, 토큰 대비 전체적인 성능과 발음 학습에 있어 가장 중요한 Phoneme Score는 ChatGPT 3.5 Turbo 모델이 가장 좋았기에 본 프로젝트에서는 ChatGPT 3.5 Turbo 모델을 채택하였다.

## ⑧ 단어 검색 기능

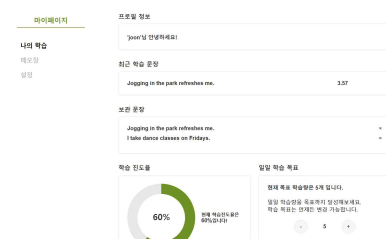
문장 추천을 기반으로 하는 영어 학습 서비스인 만큼, 사용자가 접하게 될 문장에는 사용자가 모르는 단어가 포함될 가능성이 높을 것이라 생각하였다. 따라서, 사용자가 단어 검색을 위해 추가적인 페이지를 열지 않고 해당 서비스 내에서 이를 해결할 수 있도록 페이지마다 '단어 검색' 기능을 포함시켰으며, 이는 URL 파라미터로 변환된 검색어를 Cambridge Dictionary 영-한 사전 페이지에 전달하고 전달된 페이지가 새 창으로 열리도록 구현하였다.



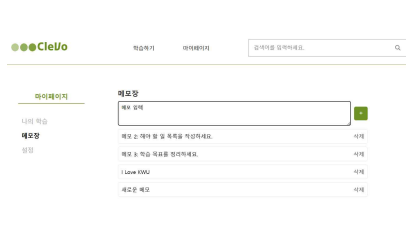
단어 검색 시 화면 ('confuse' 검색)

## ⑨ 마이페이지 기능

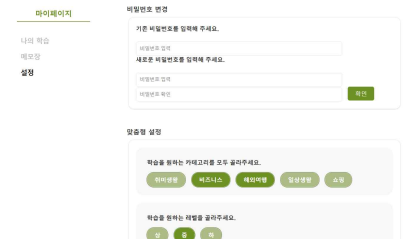
마이페이지에서는 비밀번호, 학습 카테고리, 학습 난이도 등의 회원 정보 변경 기능 이외에도 개인별 메모장 기능과 최근 학습 문장, 보관 문장, 데일리 학습 진도율 관리 등의 기능을 DB에 기록된 정보들을 바탕으로 제공한다.



<문장 보관, 학습로그, 진도율>



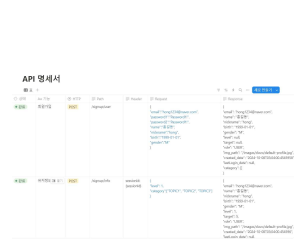
<메모장>



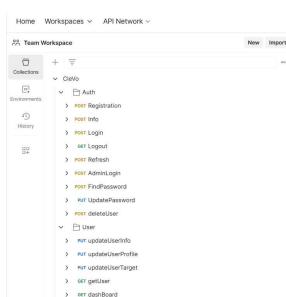
<정보 수정 페이지>

## ⑩ 실행 기능 API 구현

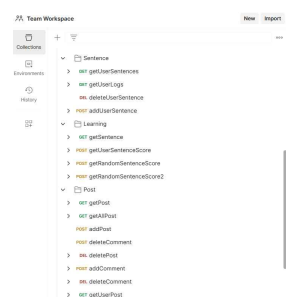
설계한 기능들이 웹서비스 상에서 작동할 수 있도록 하는 다양한 기능들은 Spring Boot 기반의 메인 서버 아래 존재하는 다양한 API들에 의해 동작하므로, 이를 위한 API들을 구현하고 관리한다. API 관리는 1차적으로 Notion에서 API 명세서를 작성하고, 최종적으로는 POSTMAN을 활용하여 관리하였다.



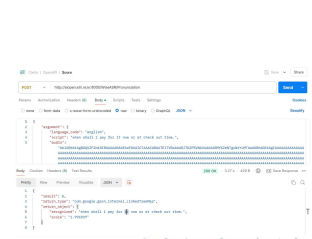
<API 명세서>



<POSTMAN API1>



<POSTMAN API2>

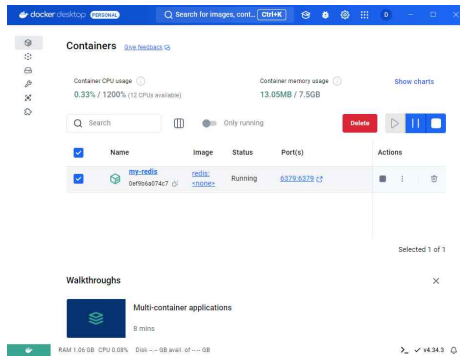


<POSTMAN API 호출 화면>

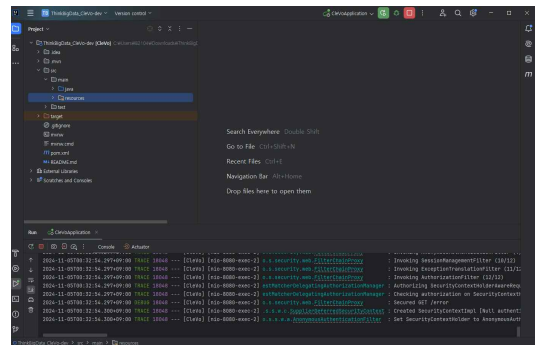
## ㉑ 서비스 배포

이렇게 구현된 웹서비스 CleVo는 SpringBoot 기반의 Local 서버와 Docker 기반의 Redis 서버를 활용해 동작하며, Local 환경에서 배포되는 것을 기본으로 한다.

이후에는 해당 프로젝트를 jar 파일로 패키징 후, AWS EC2 서버에 업로드한 다음, EC2에서 Redis 서버와 jar 파일을 실행시켜 SpringBoot 서버를 모두 작동시키는 방법으로 해당 서비스를 AWS 환경에서 배포한다.



Redis 서버 실행 모습



SpringBoot 서버 실행 모습

## 3) 개발의 내용

### 가. 수행 방법 및 추진 과정

#### 역할 분담

씹크빅데이터 (CleVo)				
오승준(팀장)	신원석	이원빈	김민서	설수빈
AI Engineer / PM	Data / BE-Engineer	Data / BE-Engineer	Designer / FE-Engineer	Designer / FE-Engineer
<ul style="list-style-type: none"> <li>Product Directing</li> <li>모델 튜닝(Prompt Engineering)</li> <li>AWS Lambda기반 모델 API제작 및 배포</li> </ul>	<ul style="list-style-type: none"> <li>Spring Boot기반 서버 관리</li> <li>서버 내 모든 API제작 및 관리</li> <li>DB관리(MySQL)</li> </ul>	<ul style="list-style-type: none"> <li>발음 평가 API관련 작업</li> <li>데이터 인코딩 및 포맷 관리</li> <li>프론트엔드-백엔드 연결 작업</li> </ul>	<ul style="list-style-type: none"> <li>js, css, html기반 페이지 제작 및 기능 구현</li> <li>(학습 기능 중심)</li> <li>서비스 디자인 &amp; 페이지 디자인 설계 및 구현</li> </ul>	<ul style="list-style-type: none"> <li>js, css, html기반 페이지 제작 및 기능 구현</li> <li>(회원 기능 중심)</li> <li>서비스 디자인 &amp; 페이지 디자인 설계 및 구현</li> </ul>



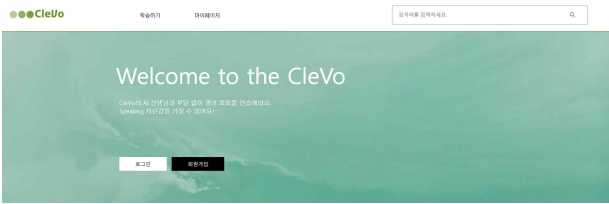


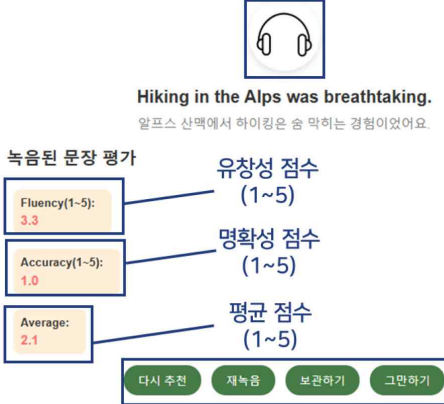
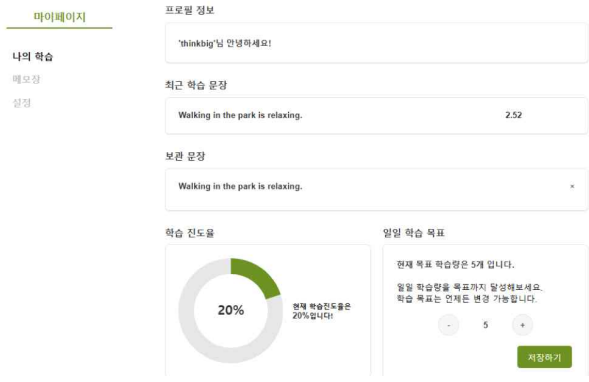
## 수행 일정

주차	내용	세부내용
4월 1~2주차	아이디어 확정 & 기업미팅	<ul style="list-style-type: none"> <li>프로젝트 아이디어 확정</li> <li>세부 계획 수립 &amp; 설계</li> <li>서비스 기능 회의</li> </ul>
4월 3~5주차		
5월 1~2주차	ERD 설계 & API 명세서	<ul style="list-style-type: none"> <li>제안된 기능에 기반한 ERD 설계</li> <li>제안된 기능에 기반한 API 명세서 제작</li> </ul>
5월 3~5주차	프로토타입 제작 1	<ul style="list-style-type: none"> <li>API를 활용한 학습하기 기능 위주의 프로토타입 제작 (핵심 기능 구현 우선)</li> </ul>
6월 1~2주차	프로토타입 제작 2	<ul style="list-style-type: none"> <li>핵심 기능 &amp; 초기 디자인을 반영한 프로토타입 제작</li> <li>중간보고서 작성</li> </ul>
6월 3~4주차		
7월 1~2주차	추천 모델 고도화	<ul style="list-style-type: none"> <li>추천 방식에 대한 고민 &amp; 모델 고도화</li> </ul>
7월 3주차	부가 기능 구현 & 미구현 기능 보완	<ul style="list-style-type: none"> <li>미구현 기능과 부가 기능들 구현</li> <li>각 파트별 마무리되지 않은 작업 수행</li> </ul>
7월 4~5주차		
8월 1주차	중간 기능 점검	<ul style="list-style-type: none"> <li>각 파트별 마무리되지 않은 작업 수행</li> <li>진행상황 공유 회의</li> </ul>
8월 2~3주차	발음 평가 (정확도 Score, 취약 발음 패턴 추출)	<ul style="list-style-type: none"> <li>정확도 Score 계산 방식 설계</li> <li>취약 발음 패턴 추출 방식 설계</li> <li>해당 기능들을 수행할 수 있도록 API 작업</li> </ul>
8월 4~5주차	추천 모델 성능 평가 방식 설계 & 성능 평가	<ul style="list-style-type: none"> <li>각 파트별 마무리되지 않은 작업 수행</li> <li>추천 모델 성능 평가 설계 &amp; 성능 평가</li> </ul>
9월 1~2주차		
9월 3~5주차	기능 점검 주간 (프롬프트)	<ul style="list-style-type: none"> <li>추천 모델의 Prompt Engineering 고도화</li> </ul>
10월 1~2주차	에러 점검 & 배포 검토	<ul style="list-style-type: none"> <li>AWS 기반 서비스 배포 검토</li> <li>산출물 내 Error 점검 및 수정</li> </ul>
10월 3~4주차	최종 산출물 정리	<ul style="list-style-type: none"> <li>최종 보고서 작성 및 서류 정리</li> <li>전시회 패널 제작</li> </ul>
10월 5주차		



## 나. 최종 결과물

모든 서비스의 실행화면은 [별첨] 항목에 첨부하였다. 아래는 서비스의 핵심 기능들을 담은 일부 실행화면과 이 화면들에서 제공하는 기능들에 대한 간단한 설명을 포함한 것이다.

메인 페이지	로그인 페이지
 <p>자주 묻는 질문 Q. CleVo는 어떤 서비스인가요? A. CleVo는 사용자의 발음을 평가하고, 정확한 발음을 개선할 수 있도록 도와주는 서비스입니다.</p>	
문장 추천 페이지	문장 녹음 페이지
<p>다음 주어진 5개의 문장은 thinkbig님께서 고른 카테고리를 기반으로 추천된 문장입니다. 다음 중 학습할 문장 하나를 골라주세요.</p> <p>Photography is a passion of mine.</p> <p>Hiking in the Alps was breathtaking.</p> <p>I play chess with my friends.</p> <p>Do you like to travel?</p> <p>Jogging in the park refreshes me.</p> <p>Re</p>	 <p>클릭 시, 원어민 발음 제공</p> <p>한글 뜻 제공</p> <p>녹음 시작 버튼</p> <p>녹음 저장 버튼</p> <p>녹음 다운로드 버튼</p>
발음 평가 페이지	마이페이지
 <p>클릭 시, 원어민 발음 제공</p> <p>Hiking in the Alps was breathtaking.</p> <p>알프스 산맥에서 하이킹은 숨 막히는 경험이었어요.</p> <p>녹음된 문장 평가</p> <p>유창성 점수 (1~5): 3.3</p> <p>정확성 점수 (1~5): 1.0</p> <p>평균 점수 (1~5): 2.1</p> <p>다시 추천</p> <p>재녹음</p> <p>보관하기</p> <p>그만하기</p>	 <p>마이페이지</p> <p>프로필 정보</p> <p>나의 학습</p> <p>배우장</p> <p>일정</p> <p>최근 학습 문장</p> <p>Walking in the park is relaxing. 2.52</p> <p>보관 문장</p> <p>Walking in the park is relaxing.</p> <p>학습 진도율</p> <p>20%</p> <p>현재 목표 학습량은 5개입니다. 일일 학습량을 목표까지 달성해보세요. 학습 목표는 언제든 변경 가능합니다.</p> <p>저장하기</p>

## 다. 소프트웨어 등록

SW등록번호	내용	
제 C-2024-040076 호	명칭	AI를 활용한 영어 발음 학습 서비스(CleVo)
	저작자	광운대학교 산학협력단

(※ 별첨 : 저작권 등록증)

## 라. 예산 집행

예산 집행 내역				
결제월	지급 항목	세부 내역	결제 금액	총액
3~5월	-	-	0 원	0 원
6월	SW 구입비	ChatGPT x 4 Google Colab Pro x 1	195,697 원	195,697 원
7월	SW 구입비	ChatGPT x 4 Google Colab Pro x 1	195,226 원	390,923 원
8월	SW 구입비	ChatGPT x 4 Google Colab Pro x 1	188,871 원	579,794 원
9월	SW 구입비	ChatGPT x 4 Google Colab Pro x 1	187,758 원	767,552 원
10월	-	-	0 원	767,552 원
총 계				767,552 원

## 마. 개선 방안

- **평가 모델의 고도화** : ETRI에서 제공하는 발음 평가 API를 사용하는 것이 아닌, AI HUB의 '한국인의 영어 발음 평가 데이터셋'을 활용하여 '정확도 점수', '유창성 점수', '취약 발음 추출'을 각각의 딥러닝 모델이 수행할 수 있도록 한 뒤, 현재보다 더 고도화된 모델을 구축하고 이를 RESTful API로 배포한다면 보다 나은 추천 성능을 기대할 수 있다.
- **추천 모델의 평가 지표와 모델 고도화** : 현재의 평가 지표는 생성된 문장의 평균적인 경향만을 기준으로 하지만, 경우에 따라서는 모든 조건을 복합적으로 만족해야 하거나, 취약 발음 우선 순위에 따라 해당 취약 발음이 얼마나 반영된 결과물이 만들어졌는지 등의 보다 고도화된 평가 기준을 설정할 수 있다. 이외에는 RAG 기법 혹은 추가적인 Prompt-Engineering 기법 적용을 통해 추천 모델의 성능을 향상시키는 방법 또한 고려해볼 수 있다.
- **부가 기능 구현** : 커뮤니티 기능이나 오늘의 문장 등의 학습 외적인 부가 기능들의 도입은 전체적인 서비스 완성도를 높일 수 있다.

### 3. 오픈소스SW 활용 및 기여

#### 가. 오픈소스SW 활용

##### 1) 활용한 오픈소스SW 소개

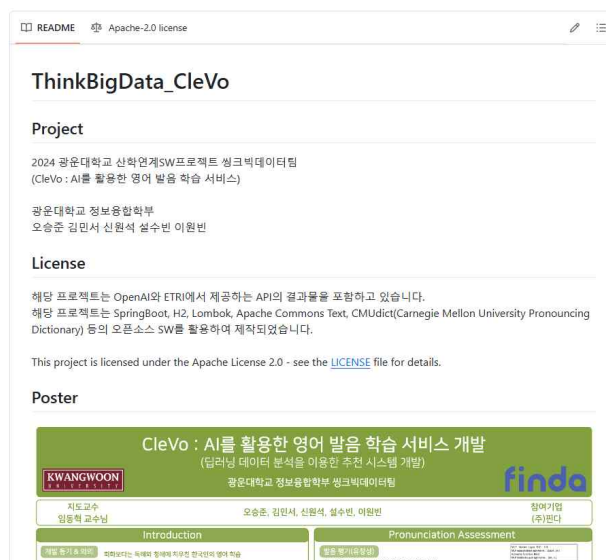
- ① Spring Boot 관련 라이브러리 (<https://github.com/spring-projects/spring-boot>)
  - spring-boot-starter-\* 패키지들: 웹 개발, 데이터 JPA, Redis, 시큐리티, OAuth2, 메일, 유효성 검증 등을 포함
  - spring-boot-devtools: 개발 중인 애플리케이션의 자동 재시작 및 빠른 피드백을 지원
  - spring-boot-starter-thymeleaf: 템플릿 엔진 Thymeleaf를 지원
  
- ② 데이터베이스 관련 라이브러리
  - com.h2database:h2: 경량 데이터베이스 H2를 지원하며, 보통 테스트 및 개발 환경에서 사용 (<https://www.h2database.com/html/main.html>)
  - com.mysql:mysql-connector-j: MySQL 데이터베이스와의 연동을 위한 JDBC 드라이버 (<https://dev.mysql.com/downloads/connector/j/>)
  
- ③ 보안 및 인증 관련 라이브러리
  - io.jsonwebtoken:jwt-\*. JSON Web Token(JWT)을 생성, 파싱 및 서명하는 라이브러리 (<https://github.com/jwtkt/jjwt>)
  
- ④ 도구 라이브러리
  - org.projectlombok:lombok: 반복적인 코드를 줄여주는 코드 생성 라이브러리 (<https://projectlombok.org/>)
  - CMUdict : 미국 영어 단어들의 발음을 표시한 사전 (<https://github.com/cmusphinx/cmuclt>)
  - org.apache.commons:commons-text: Apache Commons의 텍스트 관련 유틸리티 기능을 제공 (<https://commons.apache.org/proper/commons-text/>)
  
- ⑤ 테스트 관련 라이브러리(Spring Boot) (<https://github.com/spring-projects/spring-boot>)
  - spring-boot-starter-test: Spring Boot 기반의 테스트 기능을 제공
  - spring-security-test: Spring Security 기능을 테스트할 때 필요한 라이브러리

## 2) 활용 내용

오픈소스 SW 활용 내용
<ul style="list-style-type: none"> <li>• SpringBoot 관련 라이브러리 : SpringBoot 기반 서버를 구축하는데 있어 중요하게 사용되었으며, 이는 단순한 서버 구현을 넘어 보안, JPA DB, Redis 서버, 피드백, 테스트 등 다양한 영역에 활용되었다.</li> <li>• 데이터베이스 관련 라이브러리 : MySQL과의 연동 및 프로젝트에 적합한 경량 DB인 H2 사용 과정에 활용되었다.</li> <li>• 보안 및 인증 관련 라이브러리 : 토큰 기반 로그인(회원 관리 기능) 구현에 있어 자주 활용되었다.</li> <li>• 도구 라이브러리(Lombok) : 플러그인으로써 반복되는 코드를 줄여, 코드의 간결성과 작업 피로도를 낮추는데 활용되었다.</li> <li>• 도구 라이브러리(CMUDict) : 정확성 점수와 취약 발음 추출 과정에서 문장들을 CMU음소로 변환하는데 활용되었다.</li> <li>• 도구 라이브러리(Apache Commons) : 라이브러리에서 제공하는 텍스트 관련 유틸리티 기능 중 '레벤슈타인 거리 계산' 기능이 있어, 정확성 점수 산출을 위한 PER 계산에 필요한 레벤슈타인 거리를 구하는데 활용되었다.</li> </ul>

## 나. 오픈소스SW 기여

CleVo의 GitHub([https://github.com/warmcotton/ThinkBigData\\_CleVo](https://github.com/warmcotton/ThinkBigData_CleVo))는 Public & Apache-2.0 라이선스를 따라 코드를 공개 배포함으로써 해당 프로젝트를 오픈소스SW로 공개하였다. 해당 프로젝트를 완성하는데 있어 다양한 오픈소스SW의 도움을 받았던 만큼 이번에는 씽크빅데이터 팀의 프로젝트 코드를 오픈소스SW로 공개하는 방법으로 오픈소스 생태계에 기여할 수 있었다.



ThinkBigData CleVo GitHub README

## 4. 과제의 향후 계획

### 가. 활용 방안

- 커뮤니티 등의 부가적인 기능들이 추가된다면 고급화된 개인 맞춤형 학습 플랫폼으로써 발전할 수 있다.
- 학교 및 교육 기관의 효율적인 영어 발음 학습 도구로서 기능할 수 있다.
- 서비스를 확장시킨다면 영어 이외에도 다양한 외국어 발음 교육 프로그램으로서 기능할 수 있다.
- 취약 발음 추출에 조금 더 집중한다면 언어 치료 및 발음 교정 클리닉 등에도 활용될 수 있다.

### 나. 기대 효과

- 영어 학습자의 파트별 숙련도 불균형 해소에 도움을 줄 수 있다.
- 일상생활 속에서도 짧은 시간으로 학습이 가능하다는 점에서 자기개발의 기회를 제공할 수 있다.
- 개인화된 학습 자료를 낮은 비용과 함께 다량으로 제공할 수 있으므로 교육 격차 해소에 기여할 수 있다.
- 많은 사용자들이 공감할 수 있고 필요성을 느꼈던 문제를 해결하기 위해 기획되었던 서비스인 만큼, 다수의 사용자에게 유용함을 제공할 수 있다.

본 프로젝트는 이러한 활용 방안과 기대 효과 아래, 'CleVo : AI를 활용한 발음 학습 서비스'라는 이름에 걸맞는 사회적 기여점을 갖는다.

## 5. 별첨

### 가. 저작권 등록증

제	C-2024-040076	호
---	---------------	---



## 저작권 등록증

1. 저작물의 제호(명칭)	AI를 활용한 영어 발음 학습 서비스(CleVo)
2. 저작물의 종류	컴퓨터프로그램저작물>응용프로그램>교육
3. 저작자 성명(법인명)	광운대학교 산학협력단 서울특별시 노원구 광운로 20 광운대학교 산학협력단 (월계동)
4. 생년월일(법인등록번호)	260171-0004099
5. 창작연월일	2024년10월22일
6. 공표연월일	-
7. 등록연월일	2024년10월29일
8. 등록사항	저작자 : 광운대학교 산학협력단, 창작 : 2024.10.22

『저작권법』 제53조에 따라 위와 같이 등록되었음을 증명합니다.

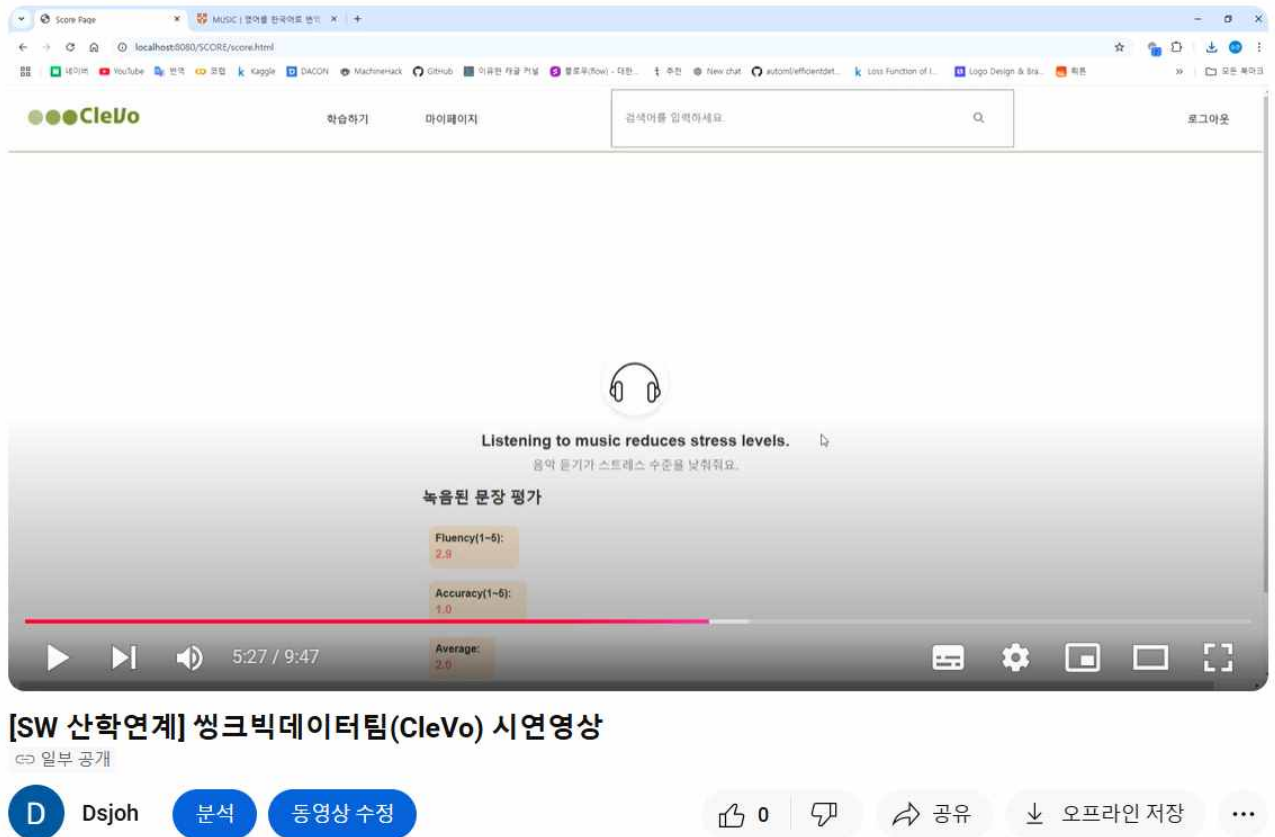
2024년 10월 30일

한국저작권위원회





## 다. 유튜브 스냅샷

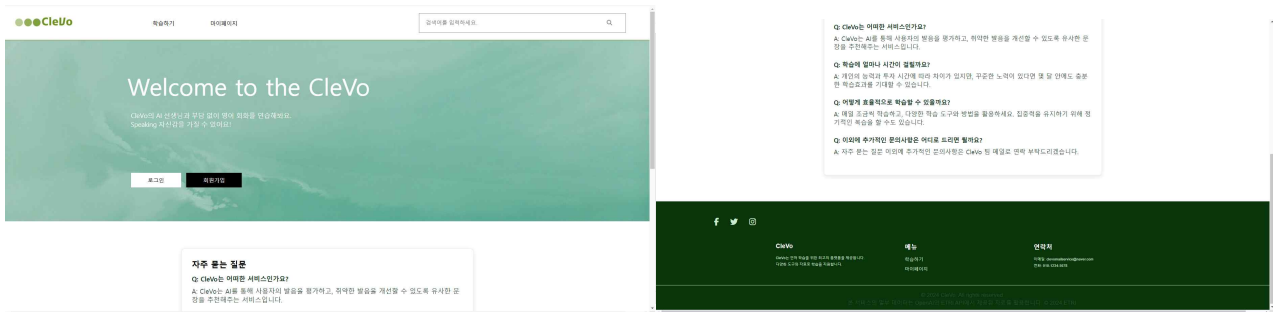


<https://youtu.be/yWXYF2xx75g>



## 라. 서비스 실행 화면

### < 메인 화면 >



### < 회원가입 >

회원가입

kw123 @ gmail.com gmail.com

.....

8~2000의 영문, 숫자, 특수문자를 모두 포함한 비밀번호를 입력해주세요

.....

비밀번호가 일치하지 않습니다

비밀번호가 일치하지 않습니다

중복 확인

사용 가능한 비밀번호가 없습니다

비밀번호

2019 10 9

남성 여성

가입하기

이미 회원이신가요? 로그인 하러가기

### < 로그인 >

CleVo

kw123@gmail.com

.....

로그인

비밀번호 찾기 | 회원가입

### < 비밀번호 찾기 >

비밀번호 찾기

kw123 @ gmail.com gmail.com

영크비

2019 10 9

재발급

### < 추천 정보 수집 (테마, 난이도) >



학습을 원하는 카테고리들 모두 골라주세요.

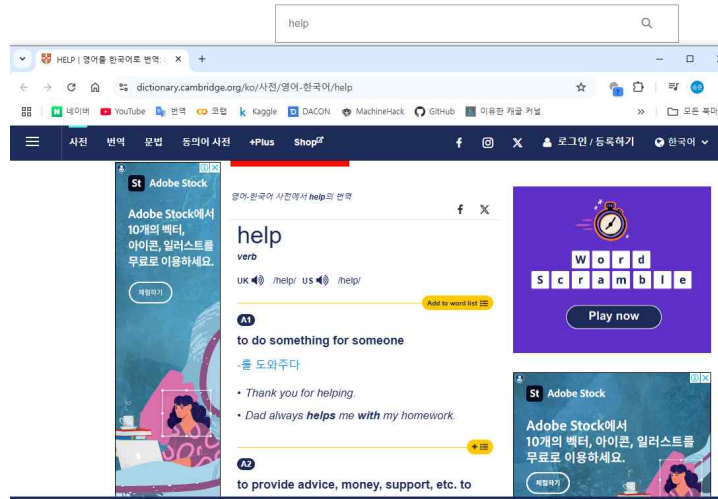
영어연습 비즈니스 해외여행 일상생활 수학

학습을 원하는 레벨을 골라주세요.

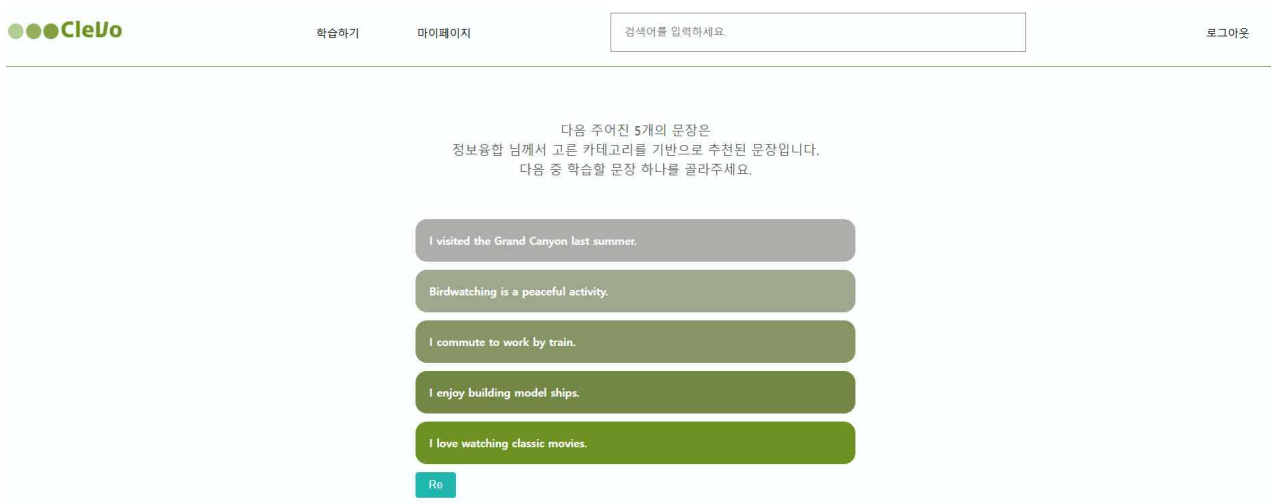
상 중 저

SUBMIT

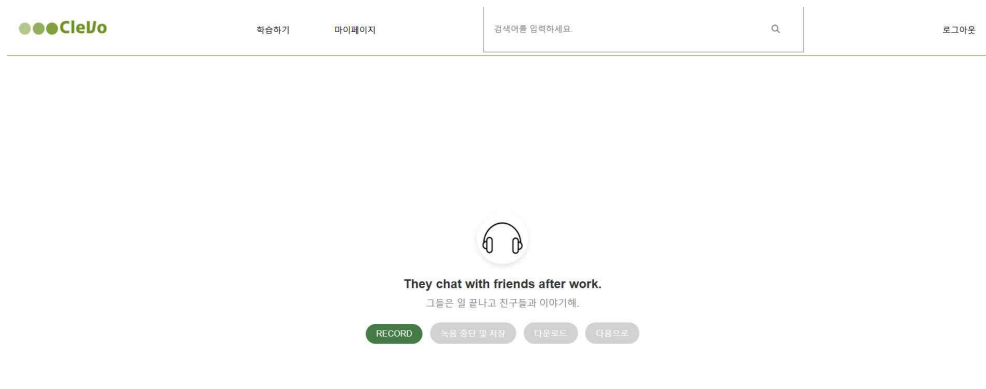
## < 단어 검색 기능 >



## < 문장 추천 >



## < 문장 선택 후 >



## < 문장 녹음 후 >



[학습하기](#) [마이페이지](#)

[로그아웃](#)



They chat with friends after work.

그들은 일 끝나고 친구들과 이야기해.

[RECORD](#)[녹음 중단 및 저장](#)[다운로드](#)[다음으로](#)

## < 녹음 파일 다운로드 >



[학습하기](#) [마이페이지](#)

[로그아웃](#)



They chat with friends after work.

그들은 일 끝나고 친구들과 이야기해.

[RECORD](#)[녹음 중단 및 저장](#)[다운로드](#)[다음으로](#)

## < 평가 결과 >



They chat with friends after work.

그들은 일 끝나고 친구들과 이야기해.

녹음된 문장 평가

Fluency(1~5):

3.9

Accuracy(1~5):

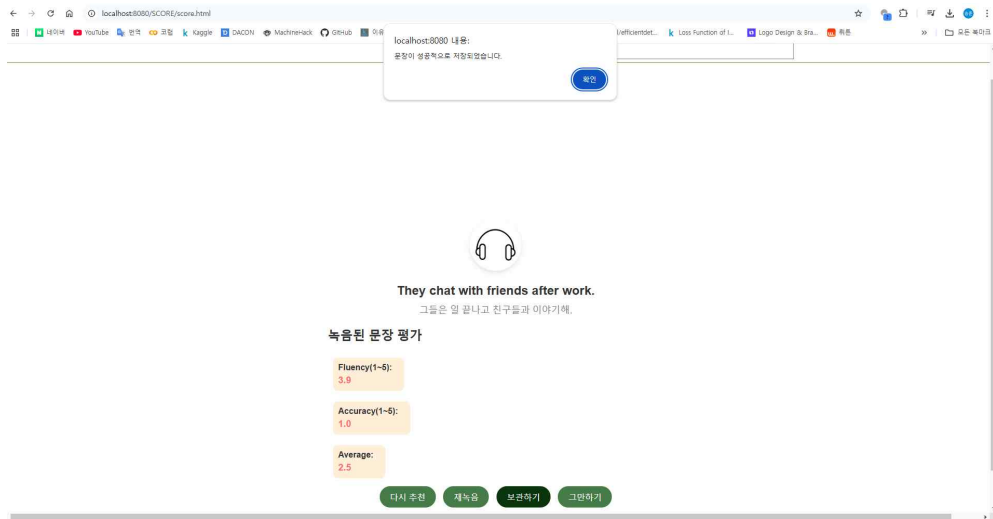
1.0

Average:

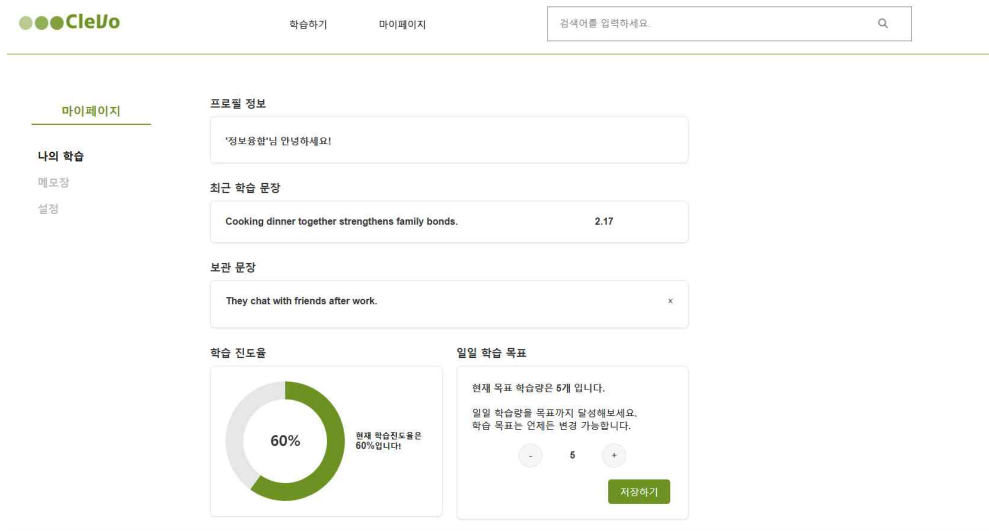
2.5

[다시 추천](#)[재녹음](#)[보관하기](#)[그만하기](#)

## < 문장 보관 시 >



## < 마이페이지 >



## < 메모장 >



## < 정보 수정 페이지 >

[학습하기](#)[마이페이지](#)

### 마이페이지

[나의 학습](#)[메모장](#)[설정](#)

#### 비밀번호 변경

기존 비밀번호를 입력해 주세요.

새로운 비밀번호를 입력해 주세요.

[확인](#)

#### 맞춤형 설정

학습을 원하는 카테고리 모두 골라주세요.

[취미생활](#)[비즈니스](#)[해외여행](#)[일상생활](#)[쇼핑](#)

학습을 원하는 레벨을 골라주세요.

[상](#)[중](#)[하](#)[저장하기](#)