

# PNU SW학습공동체 중간보고서

후견지명

## 1. 프로젝트 소개

### 가. 배경 및 필요성

오픽 시험은 시험을 치기 전에 모든 개인이 서베이를 통해 자신의 관심 분야를 고른다. 그 후 시험의 거의 모든 문제가 본인이 선택한 background, 난이도, survery의 결과에 따라 문제가 나오게 된다. 문제는 오픽의 이러한 방식이 시험을 준비하는 학생들의 입장에서 모의고사로 연습해보기 쉽지 않다는 것이다. 시중에 나와있는 무료 모의고사의 경우 내가 절대로 선택하지 않을, 이를 테면 악기 연주, 미술 감상, 수영 등 나의 취미나 관심사가 아님에도 불구하고 문제가 나오기 때문에 완벽한 준비를 하기에 어렵다. 이를 해결하고자 개인 맞춤형 오픽 모의고사 시스템 개발을 다짐하게 되었다.

### 나. 개발목표 및 주요내용, 세부내용 등

#### ▶ 개발목표 및 주요내용

- 1) 서베이와 난이도 설정에 따른 개인 맞춤형 문제 제작
- 2) 목소리 녹음 기능
- 3) 녹음된 목소리를 스크립트로 변환하고 자동 첨삭

#### ▶ 세부내용

##### [요구사항]

- 1) 기존 OPIC 기출문제를 대체할 수 있는 데이터

오픈소스, API로 얻어낼 수 있는 OPIC 기출문제를 수집한 다음, 실제 시험과 비슷한지 확인 과정이 필요하다.

- 2) 사용자가 만든 답변을 음성적으로 분석할 수 있는 시스템 구현

LLM API를 활용하여 사용자의 답변에 대한 음성 분석을 진행한다. 프롬프트를 조정하여 답변 내용을 이해가 쉽도록 요약하고, 필요시 finetuning을 통해서 성능을 개선시킬 수 있도록 한다.

### 3) 사용자의 실력 개선을 직관적으로 요약해서 설명해줄 수 있는 시스템 구현

사용자의 답변 수준을 수치화할 수 있도록 조정하고, 나온 수치 변화를 그래프로 요약하여 보여줄 수 있도록 한다.

#### [제한사항/대안]

##### OPIC 기출문제 접근 제한

- 직접적으로 OPIC의 기출문제를 접근해서 사용할 수 있는 방법이 없는 상황이다.
- 대안으로 LLM을 활용하여 OPIC 문제를 생성하는 방식이 있는데, 생성된 문제를 수집하고, 문제들의 퀄리티가 실제 기출문제와 비슷한 형식을 가지는지 검증하는 과정을 꼭 거쳐야 한다.
- 피드백은 실제 OPIC 시험 경험자를 대상으로 받을 수 있도록 할 것이다.

#### [개발 환경]

개발은 각자 로컬 노트북으로 진행할 계획이다. API만 사용한다고 가정하였을 때 gpu 리소스가 많이 필요한 상황은 아닐 것이다. 하지만 학습이 필요해지면 colab같은 클라우드 서비스들을 활용해 학습을 진행할 계획이다.

## 다. 사회적가치 도입 계획 등

현재 OPIC 평가 및 분석 서비스를 제공하고 있는 INEAR는 1회 이용권 11,000원이라는 높은 가격이 책정되어 있다. 이번 프로젝트는 이와 다르게 낮은 가격으로 차별성을 줄 것이다. API를 활용한 효율적인 시스템 구성을 통해 비슷한 수준의 서비스를 낮은 가격으로 제공할 수 있다면 경쟁력을 가질 수 있을 것으로 생각된다. 구독제로 서비스를 운영하여서 사용자의 꾸준한 참여를 유도하고, OPIC 시험 이후에도 회화 연습 수단으로써 해당 서비스가 이용될 수 있도록 성장시킨다면 효율적으로 서비스가 사용될 수 있을 것이다. 이렇게 본 시스템은 고가의 유료 서비스인 INEAR와 차별화하여 저렴한 구독 모델로 제공함으로써, 대학생 및 지역 사회의 영어 회화 수요를 만족시키는 동시에 지속적인 회화 연습 도구로 확장 가능성을 가진다.

## 2. 진행상황

### 가. 오픽 분석 및 문제 자료 수집

#### 1) 오픽(OPIc) 시험 개요

오픽(OPIc)은 "Oral Proficiency Interview - computer"의 약자로, 영어 회화 능력을 평가하는 컴퓨터 기반 시험입니다. 이 시험의 목적은 응시자가 실제 상황에서 얼마나 자연스럽게 효과적으로 영어를 구사할 수 있는지를 평가하는 것입니다. 오픽 시험은 문법, 발음, 유창성 등 기본적인 언어 능력을 평가하면서, 실제 사용 능력을 중심으로 평가합니다. 오픽 시험은 주로 서베이를 통해 응시자의 성향과 선호도를 파악하고, 그에 맞는 질문들을 제공하는 방식으로 진행됩니다. 이를 통해 각 응시자가 평소에 자주 겪거나 관심 있는 주제에 대해 대답하도록 유도하여 더 자연스러운 회화 능력을 평가할 수 있습니다.

#### 2) 오픽 시험의 서베이 및 진행 순서

##### [서베이]

시험 시작 전, 응시자는 간단한 설문조사에 참여하여 자신의 성향, 경험, 관심사 등을 선택하게 됩니다. 서베이 항목은 여행, 취미, 직장 생활 등 다양한 일상적인 주제들로 구성되어 있으며, 응시자는 자신이 익숙하거나 관심 있는 주제를 선택하게 됩니다. 이 서베이 결과를 바탕으로, 각 응시자에게 맞춤형 문제가 출제됩니다.

##### [진행 순서]

1. 서베이 응답: 응시자가 본인의 성향을 묻는 질문에 답합니다.
2. 문제 출제: 서베이 결과에 따라 응시자에게 적합한 문제들이 자동으로 출제됩니다.
3. 문제 응답: 주어진 시간 내에 각 문제에 대한 응답을 말로 제시합니다.
4. 평가: 응답 내용에 대해 점수화되며, 주로 발음, 문법, 유창성, 내용의 적합성 등을 기준으로 평가됩니다.

시험은 총 4~5개의 문제로 구성되며, 각 문제는 점차 난이도가 증가하는 형태로 출제됩니다. 문제의 난이도는 서베이 결과와 관련된 주제에 따라 다르게 설정됩니다.

#### 3) 문제 예시 및 난이도별 질문

오픽 문제는 서베이 응답에 따라 주제별로 출제되며, 각 주제에 따라 난이도가 달라집니다. 예를 들어, 여행을 주제로 한 문제에서는 주제의 난이도에 따라 다음과 같은 질문이 나올 수 있습니다:

## ■ 여행 주제 예시

- 쉬운 난이도:
  - "최근에 다녀온 여행지에 대해 이야기해 주세요."
  - "여행을 좋아하는 이유는 무엇인가요?"
- 중간 난이도:
  - "가장 기억에 남는 여행 경험을 자세히 설명해 주세요."
  - "당신이 여행을 계획할 때 중요한 요소는 무엇인가요?"
- 어려운 난이도:
  - "여행을 갈 때의 준비 과정에 대해 설명하고, 그 경험을 어떻게 관리했는지 이야기해 주세요."
  - "만약 예산이 제한된다면, 어떤 여행지를 선택하고 그 이유는 무엇인가요?"

## ■ 직장 생활 주제 예시

- 쉬운 난이도:
  - "현재 맡고 있는 직무에 대해 간단히 설명해 주세요."
  - "직장에서 하는 일 중 가장 즐거운 일은 무엇인가요?"
- 중간 난이도:
  - "최근에 직장에서 겪었던 도전적인 상황에 대해 이야기해 주세요."
  - "팀워크가 중요한 업무를 맡았을 때, 어떤 방식으로 협력했나요?"
- 어려운 난이도:
  - "최근 직장에서 맡은 중요한 프로젝트에 대해 설명하고, 그 프로젝트에서의 성과와 어려움을 공유해 주세요."
  - "직장에서의 리더십 경험에 대해 이야기하고, 어떻게 팀을 이끌었는지 설명해 주세요."

## 나. 프론트엔드

프론트엔드 개발을 맡아 React 기반의 웹 애플리케이션 구현을 준비하고 있으며, 현재는 기초적인 학습과 초기 UI 구성 작업을 진행 중이다. React의 기본 개념을 이해하기 위해 컴포넌트 기반 개발 방식과 함수형 컴포넌트 구조를 학습하였으며, 사용자 인터페이스의 동작을 제어하기 위해 `useState`, `useEffect`와 같은 기본적인 훅(Hook)의 사용법을 익혔다. 이를 바탕으로 사용자 입력에 반응하는 동적인 UI를 구현하는 실습을 진행하였다. 또한 화면 전환 기능을 위해 `React Router`의 사용법을 학습하고, 기본적인 라우팅 구조를 구성하였다.

학습 내용을 실제 프로젝트에 적용하기 위해 설문 입력 화면과 사용자 맞춤형 문제 출력 화면의 UI 초안을 구현하였다. 각 화면은 컴포넌트 단위로 분리하여 유지보수성과 확장

성을 고려한 구조로 설계하였다. 사용자 입력값을 상태로 관리하고, 이에 따라 동적으로 콘텐츠가 렌더링되도록 기본 로직을 구현하였다. 또한 향후 음성 녹음 기능과 자동 첨삭 결과와의 연동을 고려하여, 백엔드 API와의 연결이 용이한 구조로 화면을 설계하였다. 앞으로는 설문 결과를 기반으로 맞춤형 문제를 생성하고 출력하는 기능을 본격적으로 개발할 계획이다.

설문 입력

이름:  희망 난이도:

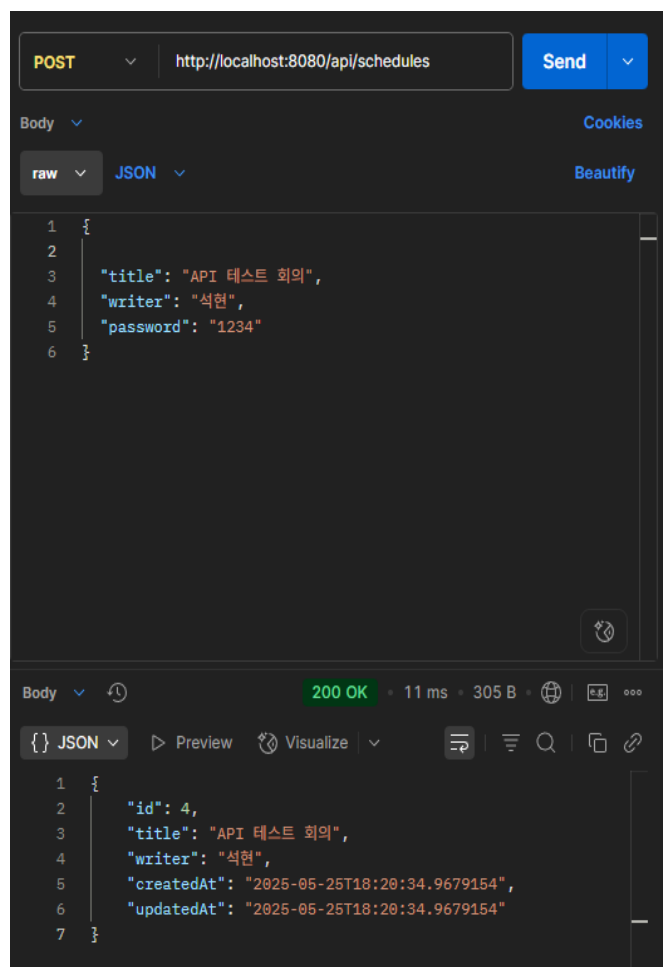
관심 분야 선택:

☒ 여행
☒ 영화
☐ 운동
☐ 음식

맞춤형 문제

- 홍길동님, IM 난이도 문제입니다.
- 관심 분야 "여행"에 대한 질문입니다.
- 관심 분야 "영화"에 대한 질문입니다.

## 다. 백엔드



백엔드 개발은 java, spring기반으로 구현을 준비하고 있다. 현재 postman을 활용한 api 사용법, 초기 구성 작업 및 학습을 진행중이다. java 사용법을 넘어서 spring의 다양한 어노테이션 활용, sql 연결 및 postman을 활용하여 api 작동과정까지 백엔드 개발에 있어서 필요한 과정들을 익히는 중에 있다. 현재 id를 기반으로 데이터베이스를 생성하고, 글쓰기 제목과 내용, 쓴 날짜까지 조회 및 수정, 삭제할 수 있는 시스템을 시험 제작해보았다. 앞으로는 사용자가 선택한 오픽 문제 특성들을 종합하고, 이에 기반하여 추천되고 생성된 문제를 사용자에게 오류 없이 잘 전달할 수 있는 기능을 개발할 계획이다. 시작하기 이전에, 효율적인 sql 구조에 대한 고려가 이루어져야 할 것이다.

[https://github.com/studyhard01/ktc\\_backend/tree/main/2nd\\_assignment](https://github.com/studyhard01/ktc_backend/tree/main/2nd_assignment)

## 라. 디자인 – CSS로 웹사이트 스타일링 학습

웹사이트 개발에서 사용자 경험(UX)과 직관적인 사용자 인터페이스(UI)를 제공하기 위해서는 시각적 요소가 필수적이다. 리액트 프로젝트 내에서 스타일을 적용하는 CSS의 역할과 특징에 대해 학습하고 실습해보았다.

### ■ CSS(Cascading Style Sheets)

- 웹 개발 시 스타일링 하기 위해 가장 대표적으로 사용되는 언어
- 웹사이트의 레이아웃과 글꼴, 색상 등의 디자인을 입히는 역할
- 하나의 스타일이 여러 엘리먼트에 적용될 수 있고, 하나의 엘리먼트에도 여러 스타일이 적용될 수 있음
- 하나의 엘리먼트가 여러 개의 스타일 규칙을 만족할 경우 해당 스타일들을 우선순위에 따라 하나씩 적용

### ■ 선택자(selector): 스타일을 어떤 엘리먼트에 적용할지를 선택하게 해줌.

#### 1) 엘리먼트 선택자: 특정 HTML 태그 이름으로 엘리먼트 선택

→ 선택자에 HTML 태그 이름 써주기

#### 2) id 선택자: 하나의 엘리먼트에 정의된 id 기반으로 엘리먼트 선택

→ 샵(#) 뒤에 아이디 넣어 사용

#### 3) 클래스 선택자: 엘리먼트의 클래스 속성으로 엘리먼트 선택

→ 점(.) 뒤에 클래스명 넣어 사용

#### 4) 전체 선택자: 전체 엘리먼트에 적용 → (\*) 사용

#### 5) 그룹 선택자: 여러 가지 선택자를 그룹으로 묶어 하나의 스타일 적용

→ 각 선택자를 콤마(,)로 구분하여 적용

#### 6) 엘리먼트의 상태와 관련된 선택자: 엘리먼트의 다양한 상태에 따라 스타일 적용

:hover – 마우스 커서가 엘리먼트 위에 올라왔을 때

:active – 주로 <a> 태그에 사용되는데 엘리먼트가 클릭됐을 때

:focus – 주로 <input> 태그에서 사용되는데 엘리먼트가 초점을 갖고 있을 경우

:checked – radio button이나 checkbox 같은 유형의 <input> 태그가 체크되어 있는 경우

:first-child / :last-child – 상위 엘리먼트를 기준으로 각각 첫 번째/마지막 child일 경우

## ■ CSS 문법과 선택자

- 선택자와 스타일로 구성됨
- 선택자 먼저 쓰고 이후에 적용할 스타일을 중괄호 안에 세미콜론(;)으로 구분하여 기술
- 각 스타일은 CSS 속성과 값으로 이뤄진 키-값 쌍이며 CSS 속성의 이름과 값을 콜론(:)으로 구분

Ex) h1 {color: green; font-size: 16px;}

h1: selector(선택자)

color, font-size: property(속성)

green, 16px: value(값)

## ■ 레이아웃과 관련된 속성

레이아웃은 화면에 엘리먼트들을 어떻게 배치할 것인지를 의미

- display: 엘리먼트를 어떻게 표시할지 그 방법에 관한 속성
- visiblity: 엘리먼트를 화면에 보여 주거나 감추기 위한 속성
- position: 엘리먼트를 어떻게 위치시킬 것인지 정의하기 위한 속성
- width/height: 가로/세로 길이를 정의하기 위한 속성
- min-width/min-height: 최소 가로/세로 길이를 정의하기 위한 속성
- max-width/max-height: 최대 가로/세로 길이를 정의하기 위한 속성

## ■ 플렉스박스

- 기존 CSS의레이아웃 사용의 불편한 부분을 개선하기 위해 등장
- 플렉스 컨테이너와 플렉스 아이টে으로 구성되며 컨테이너는 여러 개의 아이টে을 포함
- 컨테이너의 플렉스와 관련된 CSS 속성은 아이টে들을 어떤 방향과 어떤 순서로 배치할 것인지를 정의
- flex-direction: 아이টে들이 어떤 방향으로 배치될 것인지를 지정하기 위한 속성
  - main axis: flex-direction으로 지정된 방향으로 향하는 축
  - cross axis: main axis를 가로지르는 방향으로 향하는 축

- align-items: 컨테이너 내에서 아이템을 어떻게 정렬할 것인지를 결정하기 위한 속성.

Cross axis를 기준으로 함.

- justify-content: 컨테이너 내에서 아이템들을 어떻게 나란히 맞출 것인지를 결정하기 위한 속성. main axis를 기준으로 함.

#### ■ 폰트와 관련된 속성

- font-family: 어떤 글꼴을 사용할 것인지를 결정하는 속성.

지정한 글꼴을 찾지 못했을 경우를 대비해서 사용할 글꼴을 순서대로 지정해 줘야 함

- font-size: 글꼴의 크기와 관련된 속성

- font-weight: 글꼴의 두께와 관련된 속성

- font-style: 글꼴의 스타일을 지정하기 위한 속성

#### ■ 많이 사용하는 기타 속성

- background-color: 엘리먼트의 배경색을 지정하기 위한 속성

- border: 엘리먼트에 테두리를 넣기 위한 속성

#### ■ styled-components

-CSS 문법을 그대로 사용하면서 결과물을 스타일링된 컴포넌트 형태로 만들어주는 오픈 소스 라이브러리

- 컴포넌트 개념을 사용하기 때문에 리액트와 궁합이 좋음

-태그드 템플릿 리터럴을 사용하여 구성 요소의 스타일을 지정



### 3. 향후계획

#### ■ 맞춤형 문제 생성 시스템 고도화

- LLM 기반 OPIC 문제 자동 생성 모델을 구축하고, 실제 기출문제와 유사한지 검증하는 실험을 진행한다.
- 사용자의 서베이 응답에 따라 적절한 난이도와 주제를 갖는 질문을 출력하는 알고리즘을 정교화한다.

#### ■ 음성 녹음 및 자동 첨삭 시스템 구현

- 프론트엔드에서 음성을 녹음하고 이를 백엔드로 전송하는 기능을 완성한다.
- 음성 인식 모델을 활용하여 녹음 내용을 텍스트로 변환하고, 이를 LLM 기반 평가 시스템과 연동하여 자동 피드백을 제공하는 기능을 개발한다.

#### ■ 사용자 실력 분석 및 시각화

- 사용자의 응답 내용 분석 결과를 점수화하고, 시간 흐름에 따른 실력 향상 추이를 시각적으로 확인할 수 있는 그래프 모듈을 추가한다.
- 성취도에 따라 개인화된 학습 피드백 및 학습 루트를 제안하는 기능을 포함시킨다.

#### ■ 프론트엔드/백엔드 연동 및 테스트

- 설문 결과, 문제 출력, 응답 제출, 피드백 확인 등의 흐름을 하나의 서비스로 통합하고, 사용자 관점에서 전체 사용 흐름을 테스트하여 오류를 수정한다.
- 사용자 피드백을 반영한 UI/UX 개선 작업도 병행한다.

#### ■ 서비스 구조 정리 및 배포 준비

- 개발한 시스템을 안정화한 후, 추후 실제 사용자들이 활용할 수 있도록 클라우드 기반 배포 및 간단한 접속 방법 문서화를 준비한다.
- 외부 평가자(오픽 응시 경험자)를 통한 베타테스트를 계획하여, 기능의 적절성 및 학습 효과성에 대한 검증을 거친다.

#### 4. 팀 소개 (소속, 구성원별 역할)

이름	소속	역할
정혜린	의생명융합공학부 데이터사이언스 전공	디자인, 문서 관리 및 작업
이명현	의생명융합공학부 데이터사이언스 전공	오픽문제 자료수집 및 총괄
하석현	의생명융합공학부 데이터사이언스 전공	백엔드 개발
백원재	의생명융합공학부 데이터사이언스 전공	프론트엔드 개발