

# 개발 프로세스 단계별 상세내용

은퇴한 각 분야 교육자들의 전문성을 활용한 AI 강의로 학생들에게 맞춤형 커리큘럼을 제공하고, 시니어들에게는 은퇴 후 새로운 직업 기회를 창출합니다.



27일까지 완료 할 일

- ☒ 유튜브 등에서 초급~고급 강의 제공하는 강의 수집 → 텍스트 변환
- ☒ 강의 스타일 및 유용성 확인



10월 1일까지 완료 할 일

- ☐ 기획서 완성
- ☐ 웹 애플리케이션 구조 완성

## 1. 자연어 처리 : 데이터셋 및 프롬프트 변환 시스템 구축

- ☐ 도메인 지식을 가진 은퇴한 교사 혹은 강사가 자연어로 본인의 지식을 입력
- ☐ 음성 / 텍스트
- ☐ 데이터를 모델 학습용 데이터셋 + 프롬프트로 적절히 변환하는 시스템
- ▼ 중요한 안건
  - ▼ 강사 혹은 교사의 스타일을 해치지 않는 선에서 프롬프트 작성
  - ▼ 학습용 데이터 셋의 컬럼 선정 및 무엇을 기준으로 데이터를 분류할 것인가

- ▼ 음성 데이터의 경우 텍스트 변환이 제대로 이루어지는지 체크 필요

## 2. 데이터 수집 및 레벨링 : 각 레벨별 데이터 수집 (데이터 세분화 및 분류)

- ☐ 예) 0~10단계의 난이도 기준에 맞춰 각 데이터를 세분화 및 그에 따른 프롬프트 생성

- ▼ 중요한 안건

- ▼ 레벨 테스트 및 해당 레벨에 맞는 데이터 셋 + 프롬프트 구축 시 난이도 기준에 대한 도메인 지식 논의 필요

- ▼ 예) 미국 유치원생 수준 ~ 대학생 수준 or 토익 0점 ~ 토익 900점 이런 느낌..?

- ▼ 난이도 구분의 기준은 강사에게 전적으로 일임

## 3. 기본 모델 구축

- ☐ 레벨에 따른 기본 모델 구축

- ☐ 각 데이터에 태그를 붙여 프롬프트 생성 시 호출하여 사용

- ▼ 중요한 안건

- ▼ 테마 혹은 레벨에 따른 태그

## 4. 강의 평가 시스템 구축

- ☐ 평가 기준 정의

- ☐ 배포 가능/불가능 판단하는 기준점

- ☐ 몇 명의 강사 및 학생 캐릭터를 설정하여 몇 가지 시나리오를 토대로 테스트

## 5. 학생용 튜터 사이트 개발

- ☐ 레벨테스트

- ☐ 레벨에 맞는 기본 모델 강의 제공

- ☐ 개인 맞춤을 위한 보완 의견 수렴하여 재학습

## 6. 강사용 어드민 사이트 개발

- ☐ 데이터 업로드 및 변환 작업

☐ 모델 학습 및 평가

☐ 실 배포 → 학생용 튜터 사이트에서 해당 모델 사용 / 학생별 개별 모델 제공

☐ 학생의 학습 진도 로그 확인 페이지

☐ 학생의 개인 맞춤을 위한 의견 수렴에 관한 페이지 및 알림

▼ 중요한 안건

▼ 데이터 업로드 및 모든 절차는 어드민 페이지를 통해서 이루어질 수 있도록

▼ 데이터셋 및 프롬프트를 강사의 검수 후 다음단계(모델 학습)가 이루어지도록

▼ 강사에게 업로드 할 데이터의 가이드라인 제공

▼ 학생이 개인 맞춤을 위한 의견 전달 시 강사에게 Alert 제공 및 시정을 위한 시스템 구축

▼ 학생의 의견에 맞게 강사가 새로운 데이터로 모델 재학습하여 학생별 맞춤 ver. 모델 제공

▼ 혹은 애초에 여러 테마?의 데이터 프롬프트 구축하여 그에 맞는 버전의 모델 제공 논의 필요

## 7. 학생별 맞춤 프롬프트 생성 및 적용 시스템 구축

☐ 학생의 개인 의견을 강사에게 전달하여 개인 맞춤으로 모델 재학습

---

## 1. 자연어 처리 프로세스

EX)



[ ebs.txt + 레벨별 데이터 분류 기준 + 레벨별 목차 데이터 ] 투입

```
⇒ { 1레벨 : [ 프롬프트 데이터 ],  
    2레벨 : [ 프롬프트 데이터 ],  
    3레벨 : [ 프롬프트 데이터 ]  
}
```

## React 프로젝트 구조

- /: 홈페이지
- /data/add: 데이터 추가 페이지
- /data/check: 데이터 검수 페이지
- /model/new: 새 모델 생성 페이지
- /model/test: 모델 테스트 페이지

### ▼ src

▼ Components : 받아온 데이터를 가공하고 렌더링 하는 부분

▼ Services : api : 서버와 통신을 담당하는 부분

▼ Pages : 페이지 화면

- useEffect로 services/api 통해 서버에 데이터를 요청하고 받아온 데이터를 컴포넌트로 넘기는 부분
- return 안에서는 해당 페이지의 전체적인 뷰를 구성하고 뷰 안에서 (데이터 등 이) 바뀌는 부분만 컴포넌트에서 부분 렌더링

▼ Utils : 컴포넌트에서 데이터 가공 중 필요한 모듈 등을 설정하는 부분

- ▼ Styles : CSS 스타일 혹은 라이브러리를 설정하는 부분
- ▼ Routes : 서브라우트 설정하는 부분 ( 실제 주소창에 입력하는 그런 라우팅 )
  - ▼ 예) <Route path="/" element={<HomePage />} /> → pages/hompage로 넘어감
- ▼ Database : 몽고DB에 커넥트 하는 부분
- ▼ App.tsx
  - ▼ 애플리케이션 최상위 컴포넌트. 전체 앱의 구조 및 전역 설정을 담당
  - ▼ routes/AppRouter에서 설정한 라우터 컴포넌트를 불러옴
- ▼ index.tsx
  - ▼ 애플리케이션 진입점. App 컴포넌트를 루트 DOM에 렌더링.

## 어드민 페이지 구조

- 대시보드 : home.tsx | homePage.tsx
  - 현재 보유한 모델, 데이터, 프롬프트 목록 및 진행상황 표시
- 데이터 업로드 페이지 : upload.tsx , createDataset.tsx , updateDataset.tsx | addData.tsx
  - 데이터 업로드
  - 레벨 별 데이터 셋+프롬프트 생성 및 저장 기능 추가
- 모델 생성 페이지 : training.tsx | newModel.tsx , updateModel.tsx
  - 저장된 데이터 셋 로드
  - 새로운 ver. 모델 생성 후 저장
  - 목록에서 알맞는 데이터셋 선택 + 저장될 모델 이름 입력 후 생성 버튼 클릭 → new model ver1으로 저장
  - 이미 만들어진 모델일 경우 불러와서 재학습
- 검수 페이지 : check.tsx , test.tsx | checkData.tsx , testModel.tsx
  - check : 업로드 및 생성한 데이터셋 검수 페이지
  - test : 모델 학습 후 테스트 실행하는 페이지
- 실 배포 버튼 : 여러 버전의 모델 중 각 레벨 별 베이스 모델은 1개씩 선택 가능

- 대시보드에서 선택 및 저장 시 해당 모델이 각 레벨 별 default 모델이 됨.

homePage.tsx : '/'

- 웹 애플리케이션 실행시 처음 보이는 페이지
- home 컴포넌트 호출

home.tsx

- 대시보드 페이지에 사용할 컴포넌트
- 현재 보유한 모델, 데이터, 프롬프트 목록 및 진행상황 표시

addData.tsx : '/addData'

- 데이터 업로드 페이지
- upload, createDataset 컴포넌트 호출
- 업로드 및 데이터셋 생성 후 화면에 생성된 데이터셋 출력
- 검수 후 업데이트 및 저장 가능

upload.tsx

- 데이터 업로드 페이지에 사용할 컴포넌트
- 데이터 업로드 및 DB 저장 기능

createDataset.tsx

- 데이터 셋 생성 컴포넌트
- 데이터 셋 및 프롬프트 생성 및 DB 저장 기능

checkData.tsx : '/checkData'

- 업로드X 생성된 데이터셋 검수 페이지
- 검수 및 DB 업데이트 기능
- updateDataset 컴포넌트 호출

updateDataset.tsx

- 데이터 셋 수정 컴포넌트
- 데이터 셋 및 프롬프트 수정 및 DB 업데이트 기능

newModel.tsx : '/newModel'

- 모델 생성 페이지

- training 컴포넌트 호출

training.tsx

- 모델 생성 컴포넌트
- 데이터셋 불러와서 모델 생성 및 DB 저장 기능

testModel.tsx : '/testModel'

- 모델 테스트 페이지
  - 테스트 방법에 대해 좀 더 논의 필요
  - 실제 튜터 사이트에 들어갈 형태의 강의 페이지를 구성하는 것이 맞나?
- test 컴포넌트 호출

test.tsx

- 모델 테스트 컴포넌트
- 모델 테스트 및 DB 저장 기능

---

## ai-tutor-studio 전체 프로젝트 구조