

Portfolio

김우석

No를 모르는
빅데이터 전문가, 김우석



김우석

생년월일

1996.03.17

학교 / 전공

고려대학교
건축사회환경공학부 전공

희망 근무지

서울

출근 가능일

2023.01

No를 모르는 빅데이터 전문가

프로그래밍 역량

Python	-----	중상
Tensorflow	-----	중상
SQL	-----	중
JavaScript	-----	중
JAVA	-----	하

경력 및 프로젝트

Webtuna (선호하는 그림체를 기반으로 웹툰을 추천하는 서비스)
백엔드 및 머신러닝, 데이터베이스 / 기여도 20% / 참여인원 6명

2022.09 - 2022.10 (6주) / SSAFY

- 웹툰 데이터 API 활용 데이터베이스 구축(부족한 부분은 크롤링)
- CNN을 활용하여 웹툰을 그림체에 따라 분류하고, 이를 기반으로 웹툰 추천
- Collaborative Filtering(Item-based Filtering & User-based Filtering) 기반 웹툰 추천
- 5가지 웹툰 추천 알고리즘 개발 및 API 구현
- AWS 활용 배포 및 AWS RDS 활용을 통한 DB 통합

#Python(Django) #Tensorflow #CNN #MySQL #AWS EC2/RDS

Young Climb (클라이밍 문제 풀이 공유 커뮤니티 어플리케이션)
백엔드 및 머신러닝, 데이터베이스 / 기여도 20% / 참여인원 6명

2022.10 - 2022.11 (6주) / SSAFY

- YOLO 알고리즘 활용 윈스팬 계산 기능 제공
- 2D to 3D GAN Model 활용 사진을 통해 클라이밍 벽 3D모델링
- MVC 패턴 이용 RESTful API 설계
- JWT활용 로그인 로직 및 API구현
- 멤버 관리 및 등급 관련 로직 및 API 구현
- 게시글 관련 로직 및 API 구현

#Java(Spring Boot) #Spring Security #Spring Data JPA #Tensorflow #CNN #GAN
#MySQL

DEVbti (MBTI를 통해 나와 어울리는 개발 직무를 찾아주는 서비스) 백엔드 및 데이터베이스 / 기여도 25% / 참여인원 4명

2022.04 – 2022.05 (4주) / 막내온탑

- 총 12가지 질문을 통해 MBTI 검사 및 어울리는 개발자 유형 추천
- 검사결과 공유를 위한 SNS 공유기능 제공(카카오톡, 페이스북, 트위터, 링크복사)
- AWS EC2 활용한 배포

[#Python\(Django\)](#) [#SQLite](#) [#HTML](#) [#CSS](#)

Recourta (음성인식과 얼굴인식을 활용한 화상 미팅 플랫폼) 프론트엔드 / 기여도 25% / 참여인원 6명

2022.07 – 2022.08 (6주) / SSAFY

- face-api 활용 수강자 얼굴 분석하여 출결 자동 관리 (대리출석방지, 자리비움확인)
- STT(Speech To Text) 기능 활용 강의 중 실시간 자막 및 강의 후 강의록제공
- 사용자 편의성을 위해 다크모드 및 모바일/PC 모두 최적화된 반응형 웹 페이지 제공

[#Vue3](#) [#JavaScript](#) [#HTML](#) [#CSS](#) [#Tailwindcss](#)

교육과정

삼성 청년 SW 아카데미 (삼성전자)

2022.01 – 2022.12

삼성전자에서 주관하는 청년 소프트웨어 인재 양성 교육 과정으로 프로젝트를 중심으로 학습 및 실습을 진행했습니다. 800시간의 코딩 집중 교육을 이수하였고, 800시간 동안 네 차례 개발 프로젝트를 진행하였습니다.



01

Webtuna

(선호하는 그림체를 기반으로 웹툰을 추천하는 서비스)

프로젝트 기간 : 2022.09 – 2022.10 (6주)

프로젝트 현황 : 베타 버전 개발 완료 및 배포 예정

Webtuna (선호하는 그림체를 기반으로 웹툰을 추천하는 서비스)

기간 : 2022.09 - 2022.10 (6주)

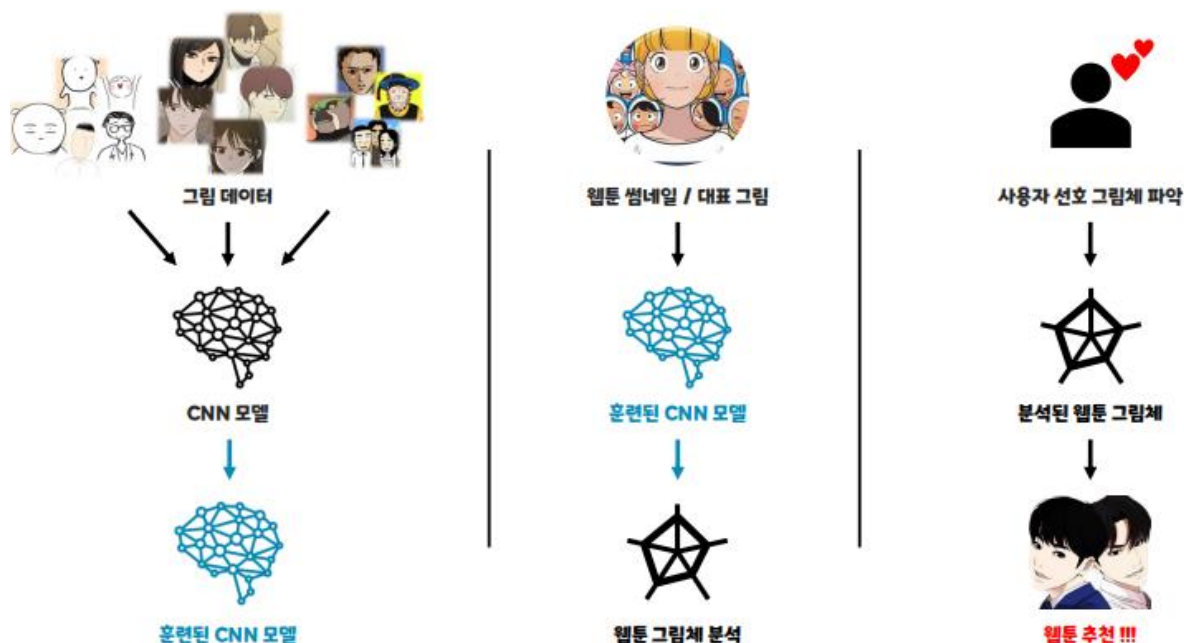
개요 : 선호하는 그림체를 기반으로 웹툰을 추천하는 서비스

CNN을 활용하여 웹툰을 분석하고, 이를 기반으로 사용자가 선호하는 그림체의 웹툰을 추천했습니다. 이외에도 5가지 추천 방식을 활용하여 웹툰 추천 서비스를 제공했습니다.

담당 역할	백엔드 및 머신러닝, 데이터베이스
기 여 도	20%
개발 환경	Python(Django), Tensorflow, CNN, MySQL, AWS EC2/RDS
구현 사항	<div><div>[서비스]</div><div>- 사용자 기반5가지 웹툰 추천 알고리즘</div><div>- 웹툰을 그림체 별 총 6개로 분류</div><div>- 사진 업로드 시 비슷한 그림체 웹툰 추천</div><div>- 웹툰 찜 목록, 보러 간 목록 구현</div><div>[개발]</div><div>- CNN(ResNet50) 활용 그림체 학습 및 분류</div><div>- RestAPI 활용 서버 클라이언트 통신</div><div>- Django와 MySQL로 DB와 서버통신</div><div>[결과]</div><div>SSAFY 2학기 특화프로젝트 1등 수상</div></div>
깃헙 주소	https://github.com/terrykim96/Webtuna

[머신 러닝]

머신 러닝을 통해 그림체 학습 후 웹툰 그림체를 분류했습니다. 사용자가 선호하는 그림체를 파악한 후 그림체를 기반으로 웹툰을 추천했습니다.



Webtuna (선호하는 그림체를 기반으로 웹툰을 추천하는 서비스)

[데이터 전처리 및 모델 생성]

6가지 그림 데이터를 전처리했습니다. 이후 ResNet50 기반의 모델을 생성했습니다.

```
[6] 1 df['class'].unique()

array(['순정st', '판타지무협', '캐릭터', '남작', '양산2d',
      '정통무협'], dtype=object)

1 df['class'].value_counts()

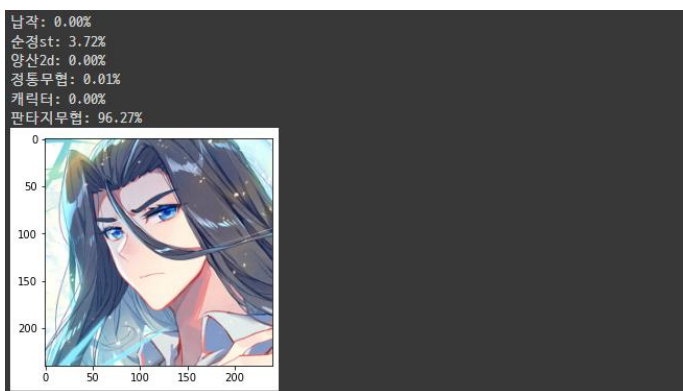
순정st      296
판타지무협  200
캐릭터      200
남작        200
양산2d      200
정통무협    200
Name: class, dtype: int64
```

```
1 def build_fit_save_cnn(input_shape, n_classes, epochs, batch_size, X_train, X_val, y_train, y_val):
2     base_model = ResNet50(weights='imagenet', include_top=False, input_shape=input_shape)
3
4     add_model = Sequential()
5     add_model.add(Flatten(input_shape=base_model.output_shape[1:]))
6     add_model.add(Dense(512, activation='relu'))
7     add_model.add(Dense(512, activation='relu'))
8     add_model.add(Dropout(0.25))
9     add_model.add(Dense(n_classes, activation='softmax'))
10
11 # combine base model and fully connected layers
12 final_model = Model(inputs=base_model.input, outputs=add_model(base_model.output))
13
14 # specify SGD optimizer parameters
15 sgd = optimizers.SGD(lr=0.001, decay=1e-6, momentum=0.9, nesterov=True)
16
17 # compile model
18 final_model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
19
20 history = final_model.fit(X_train, y_train, batch_size=batch_size, epochs=epochs,
21                          verbose=1, validation_data=(X_val, y_val))
22
23 score = final_model.evaluate(X_val, y_val, verbose=0)
24 print('Val. score:', score[0])
25 print('Val. accuracy:', score[1])
26
27 #plot_model(final_model, to_file='gdrive/MyDrive/Data/CNN-ResNet50-Kaggle-Model.svg')
28 show_history(history)
29 save_model(final_model, history)
30
31 return final_model
```

[모델 훈련 및 테스트]

모델을 훈련하고 테스트했습니다.

```
Epoch 1/15
30/30 [=====] - 26s 645ms/step - loss: 1.0099 - acc: 0.6161 - val_loss: 0.5281 - val_acc: 0.7679
Epoch 2/15
30/30 [=====] - 12s 386ms/step - loss: 0.2429 - acc: 0.9120 - val_loss: 0.2910 - val_acc: 0.9179
Epoch 3/15
30/30 [=====] - 12s 418ms/step - loss: 0.0980 - acc: 0.9648 - val_loss: 0.1212 - val_acc: 0.9571
Epoch 4/15
30/30 [=====] - 12s 486ms/step - loss: 0.0300 - acc: 0.9908 - val_loss: 0.1012 - val_acc: 0.9571
Epoch 5/15
30/30 [=====] - 13s 436ms/step - loss: 0.0297 - acc: 0.9916 - val_loss: 0.2250 - val_acc: 0.9250
Epoch 6/15
30/30 [=====] - 13s 424ms/step - loss: 0.0493 - acc: 0.9832 - val_loss: 0.1304 - val_acc: 0.9464
Epoch 7/15
30/30 [=====] - 13s 420ms/step - loss: 0.0182 - acc: 0.9958 - val_loss: 0.0960 - val_acc: 0.9714
Epoch 8/15
30/30 [=====] - 12s 418ms/step - loss: 0.0057 - acc: 0.9992 - val_loss: 0.0701 - val_acc: 0.9786
Epoch 9/15
30/30 [=====] - 13s 420ms/step - loss: 0.0034 - acc: 1.0000 - val_loss: 0.0592 - val_acc: 0.9821
Epoch 10/15
30/30 [=====] - 13s 421ms/step - loss: 0.0018 - acc: 1.0000 - val_loss: 0.0327 - val_acc: 0.9857
Epoch 11/15
30/30 [=====] - 12s 399ms/step - loss: 4.7400e-04 - acc: 1.0000 - val_loss: 0.0375 - val_acc: 0.9821
Epoch 12/15
30/30 [=====] - 12s 400ms/step - loss: 2.5581e-04 - acc: 1.0000 - val_loss: 0.0378 - val_acc: 0.9821
Epoch 13/15
30/30 [=====] - 13s 425ms/step - loss: 2.0279e-04 - acc: 1.0000 - val_loss: 0.0350 - val_acc: 0.9857
Epoch 14/15
30/30 [=====] - 13s 421ms/step - loss: 1.7910e-04 - acc: 1.0000 - val_loss: 0.0384 - val_acc: 0.9857
Epoch 15/15
30/30 [=====] - 12s 419ms/step - loss: 1.4973e-04 - acc: 1.0000 - val_loss: 0.0350 - val_acc: 0.9857
```



[ERD]

데이터 베이스 설계를 위한 ERD를 작성했습니다.



YOUNG CLIMB

CLIMBER COMMUNITY



02

Young Climb

(클라이밍 문제 풀이 공유 커뮤니티 어플리케이션)

프로젝트 기간 : 2022.10 – 2022.11 (6주)

프로젝트 현황 : 베타 버전 apk 개발 완료 및 Google Play Store 배포 준비 중

Young Climb (클라이밍 문제 풀이 공유 커뮤니티 어플리케이션)

기간 : 2022.10 - 2022.11 (6주)

개요 : 클라이밍 문제 풀이 공유 커뮤니티 어플리케이션

실내 클라이밍장 유저들이 자신이 푼 문제영상을 업로드하거나, 다른사람들의 문제풀이 영상을 검색할 수 있는 클라이밍 커뮤니티 SNS를 Spring Boot와 React Native를 활용하여 개발했습니다.

담당 역할	백엔드 및 머신러닝, 데이터베이스
기 여 도	20%
개발 환경	Java(Spring Boot), Spring Security, Spring Data JPA, Tensorflow, CNN, GAN, MySQL
구 현 사 항	<div>[서비스]<ul style="list-style-type: none">- 클라이밍 영상 업로드 시 카테고리화- 문제 검색 시 카테고리로 검색하여 문제 특정 가능- 문제 검색 시 '나와 체형이 비슷한 클라이머 먼저보기' 기능 제공- 사진 업로드 시 AI를 활용한 윈스팬 계산 기능 제공- 클라이밍장 실제 벽을 3D 모델링하여 제공<div>[개발]<ul style="list-style-type: none">- YOLO Object Detection 활용 윈스팬 계산 및 제공- 2D to 3D GAN 활용 클라이밍 벽 3D 모델링- RestAPI 활용 서버 클라이언트 통신</div><div>[결과]<ul style="list-style-type: none">SSAFY 2학기 자율프로젝트 1등 수상</div></div>
깃헙 주소	https://github.com/terrykim96/YoungClimb

Young Climb (클라이밍 문제 풀이 공유 커뮤니티 어플리케이션)

[YOLO Object Detection]

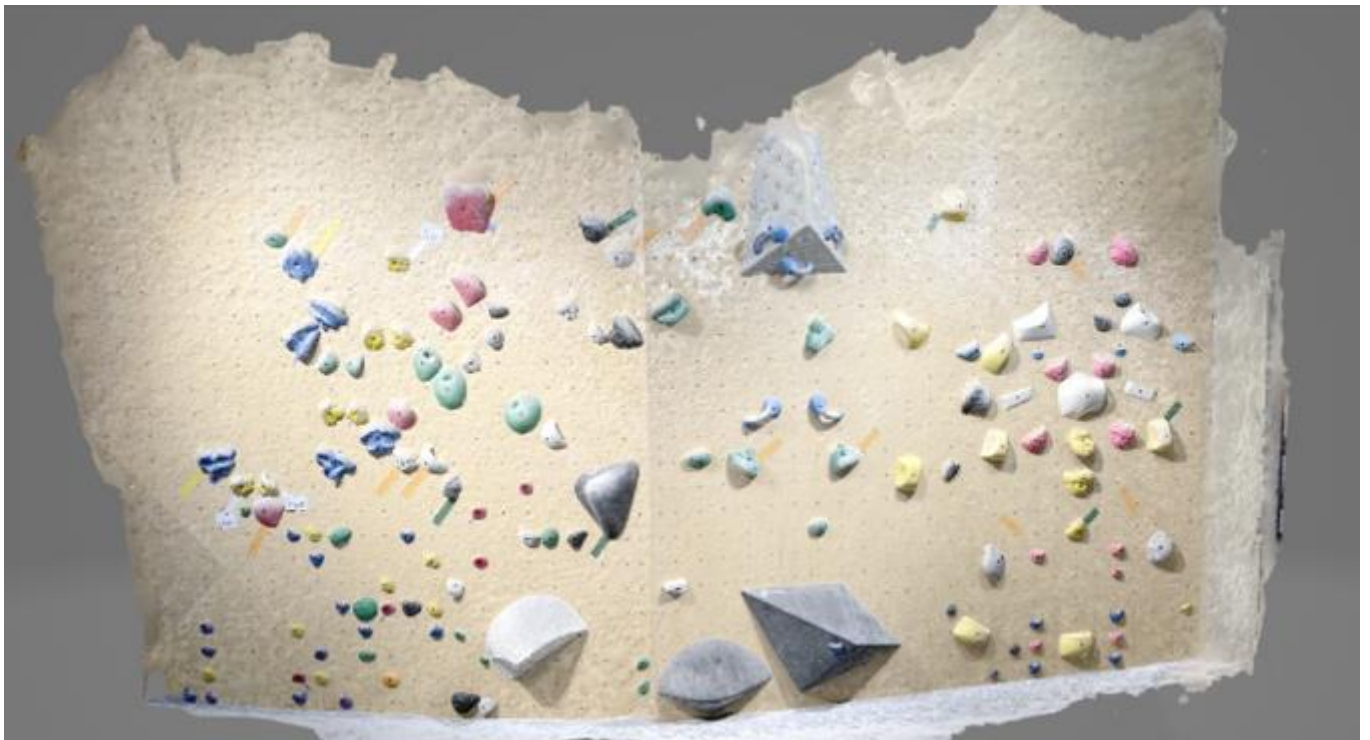
YOLO Object Detection을 활용해 사용자가 업로드한 사진에서 사람을 찾고, 비례식을 활용해 키로부터 wingspan을 계산했습니다.

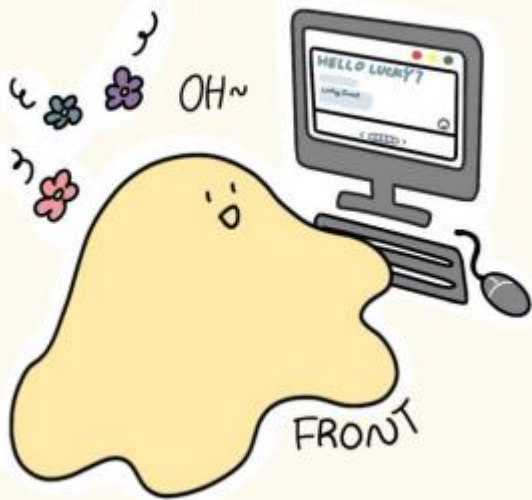


키 : wingspan = Box 세로 길이 : Box 가로 길이

[2D to 3D GAN]

2D to 3D GAN을 활용해 2D 사진으로부터 3D 벽을 모델링했습니다.





03

DEVbti

(MBTI를 통해 나와 어울리는 개발 직무를 찾아주는 서비스)

<https://devbti.com/>

프로젝트 기간 : 2022.04 - 2022.05 (4주)

프로젝트 현황 : 공식 서비스 중 - 4320명 이용
(2022.11.29 기준)

DEVbti (MBTI를 통해 나와 어울리는 개발 직무를 찾아주는 서비스)

기간 : 2022.04 - 2022.05 (4주)

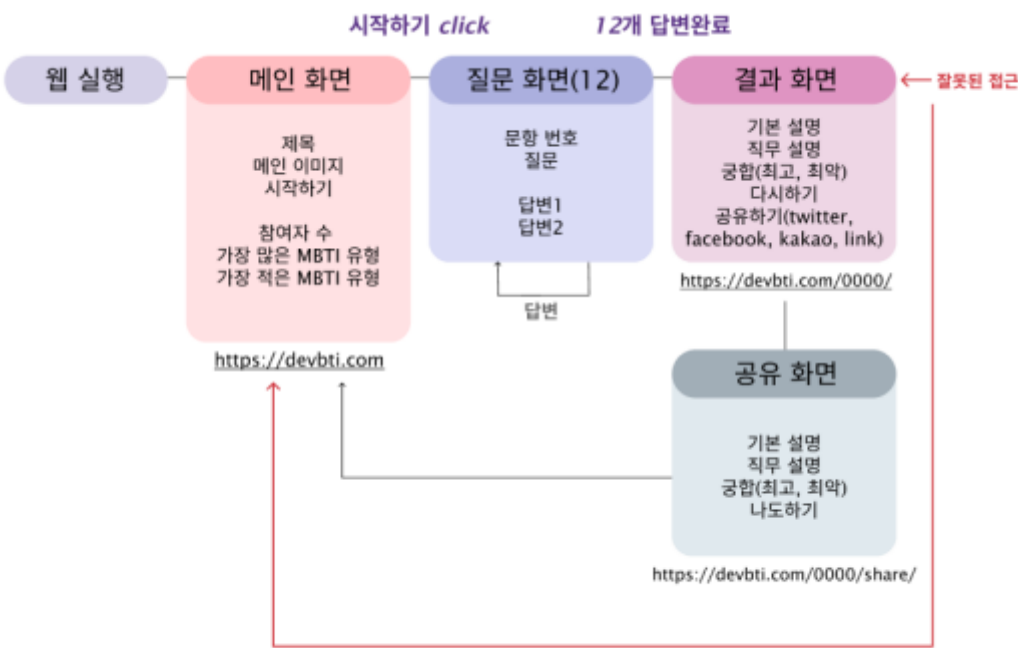
개요 : MBTI를 통해 나와 어울리는 개발 직무를 찾아주는 서비스

MBTI 기반 어울리는 개발 직무를 찾아주는 서비스를 Django와 SQLite를 기반으로 구현했습니다.

담당 역할	백엔드 및 데이터베이스
기 여 도	25%
개 발 환 경	Python(Djaingo), SQLite, HTML, CSS
구 현 사 항	[서비스] - 12개 문항으로 이루어진 개발자를 위한 MBTI 테스트 [개발] -Django MTV 패턴에 기반한 화면 템플릿 개발 -외부API를 활용한 SNS 공유 기능구현 [결과] AWS EC2를 활용한 수동 배포진행, 배포 후4000명 이상 이용
사이트 주소	https://devbti.com/

[사용자 시나리오]

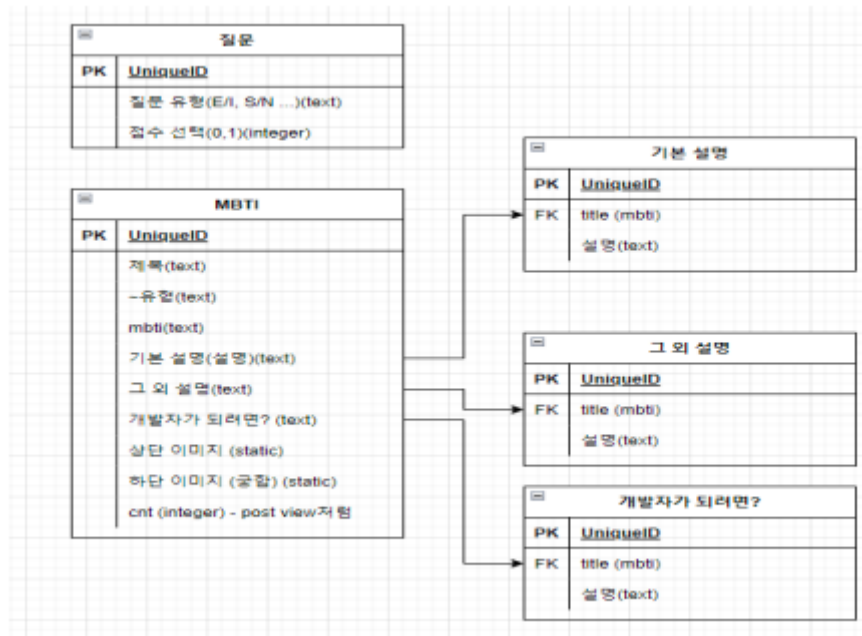
사용자 시나리오 작성을 통해 서비스 구성을 문서화하고 비정상 요청 등 세부 요청을 처리했습니다.



DEVbti (MBTI를 통해 나와 어울리는 개발 직무를 찾아주는 서비스)

[ERD]

데이터 베이스 설계를 위한 ERD를 작성했습니다.



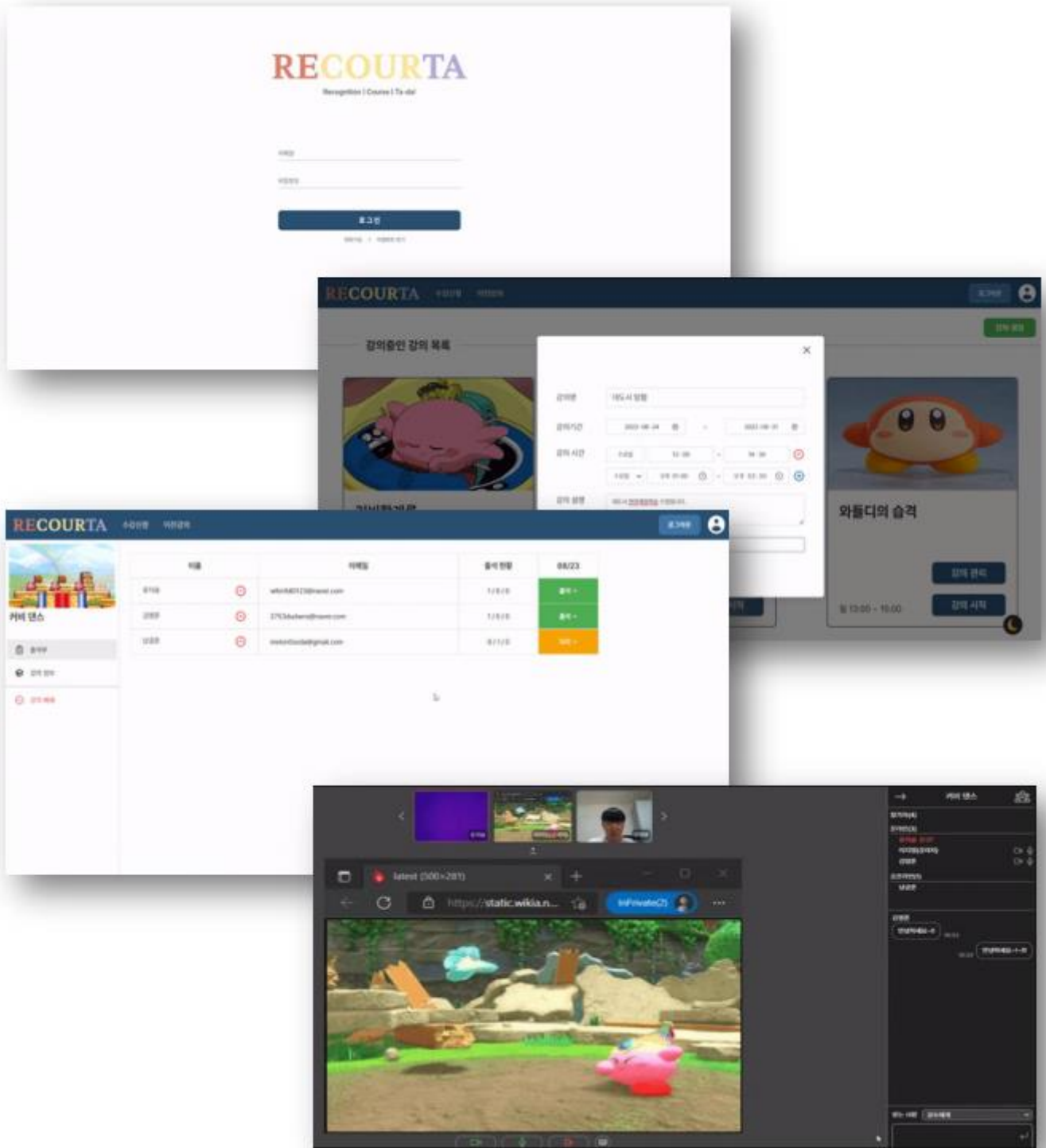
[로컬 스토리지]

12가지 질문 답변 결과 저장을 위하여 로컬 스토리지를 사용했습니다.

```
<script>
  const loadMbti = JSON.parse(localStorage.getItem('mbtiDict'))
  const loadList = JSON.parse(localStorage.getItem('testList'))
  J

  const LeftSelect = document.querySelector(`#choiceLeft`)
  LeftSelect.addEventListener('submit', onLeftSubmit)

  const RightSelect = document.querySelector(`#choiceRight`)
  RightSelect.addEventListener('submit', onRightSubmit)
</script>
```



04 Recourta

(음성인식과 얼굴인식을 활용한 화상 미팅 플랫폼)

프로젝트 기간 : 2022.07 - 2022.08 (6주)

프로젝트 현황 : 베타 버전 개발 완료

Recourta (음성인식과 얼굴인식을 활용한 화상 미팅 플랫폼)

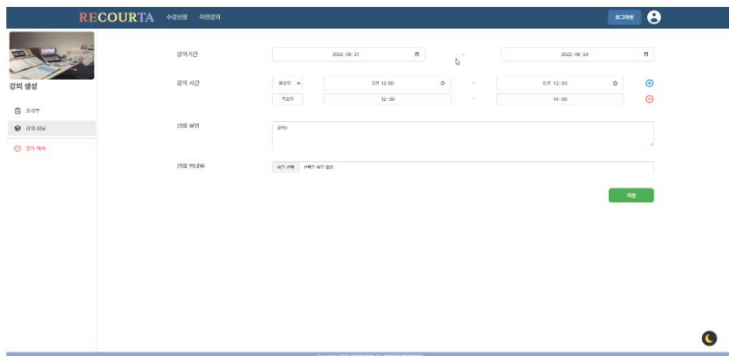
기간 : 2022.07 - 2022.08 (6주)

개요 : 음성인식과 얼굴인식을 활용한 화상 미팅 플랫폼

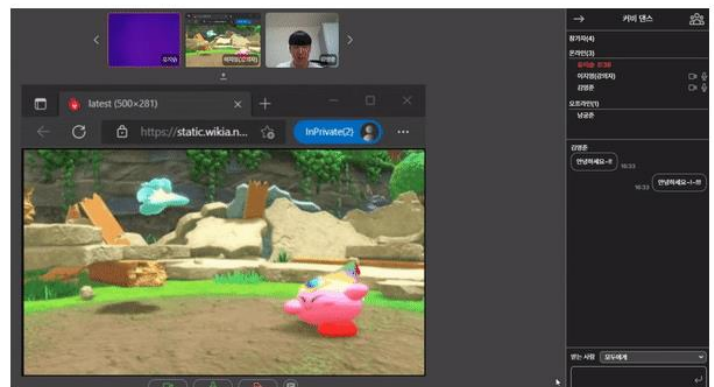
WebRTC 기능을 활용하여 온라인 화상 강의실 플랫폼을 제공했습니다.

담당 역할	프론트엔드 개발
기여도	25%
개발 환경	Vue3, JavaScript, HTML, CSS, Tailwindcss
구현 사항	<div><div>[서비스]</div><div>- 얼굴 인식을 활용한 자동 출결 인식</div><div>- 음성 인식을 활용한 실시간 강의 자막 및 강의록 제공</div><div>- 강의 생성 및 수강생 관리</div><div>[개발]</div><div>- WebRTC 활용 통신</div><div>- RestAPI 활용(Axios) 서버 클라이언트 통신</div></div>

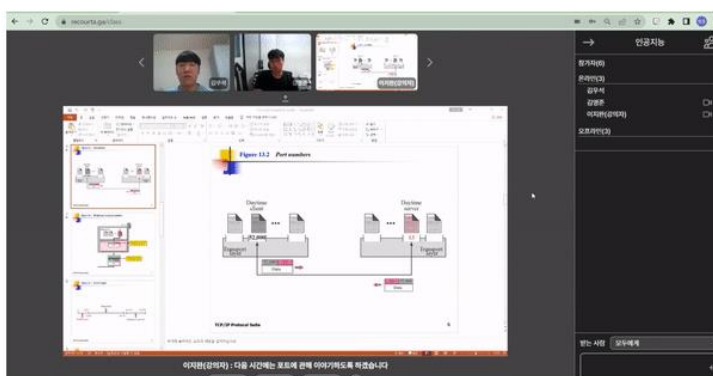
강의 정보 수정



채팅



자막



자리 비움

