



대덕인재개발원 2023년 8월반 최종프로젝트

AI기술을 활용한 소프트웨어 엔지니어링 과정

#개발자를 위한 프로젝트 통합관리 시스템

NAVI MINE



Project Management System



TEAM 3

[김태은] [우민규] [유길상] [이상철] [정민지]



#목차

AI기술을 활용한 소프트웨어 엔지니어링 과정



최종프로젝트 발표 진행순서 소개



팀 소개

프로젝트 개요

수행 절차

산출물

시연

Q & A



#팀 소개

#SKJ #삼조국비지원



● 우민규

★ BA

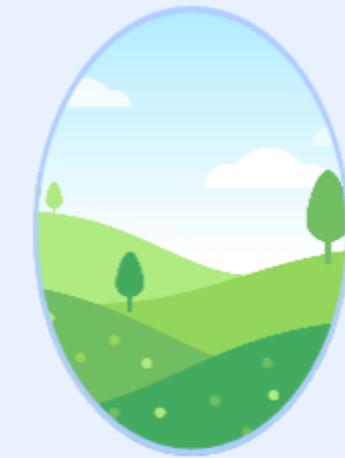
● 이곳에 역할을 입력해 주세요.



● 이상철

★ AA

● 이곳에 역할을 입력해 주세요.



● 김태은

★ PL/UA

● 이곳에 역할을 입력해 주세요.



● 유길상

★ TA

● 이곳에 역할을 입력해 주세요.



● 정민지

★ DA

● 이곳에 역할을 입력해 주세요.

AI기술을 활용한 소프트웨어 엔지니어링 과정



02 프로젝트 개요

T E A M 3 N A V I M I N E P M S



#프로젝트 개요

AI기술을 활용한 소프트웨어 엔지니어링 과정

#주제 선정 배경

4.7(조) 달러

세계 IT시장
지속 성장

103(조) 원

국내 시장 규모

5(%) 프로

국내 IT시장
2024 성장 전망



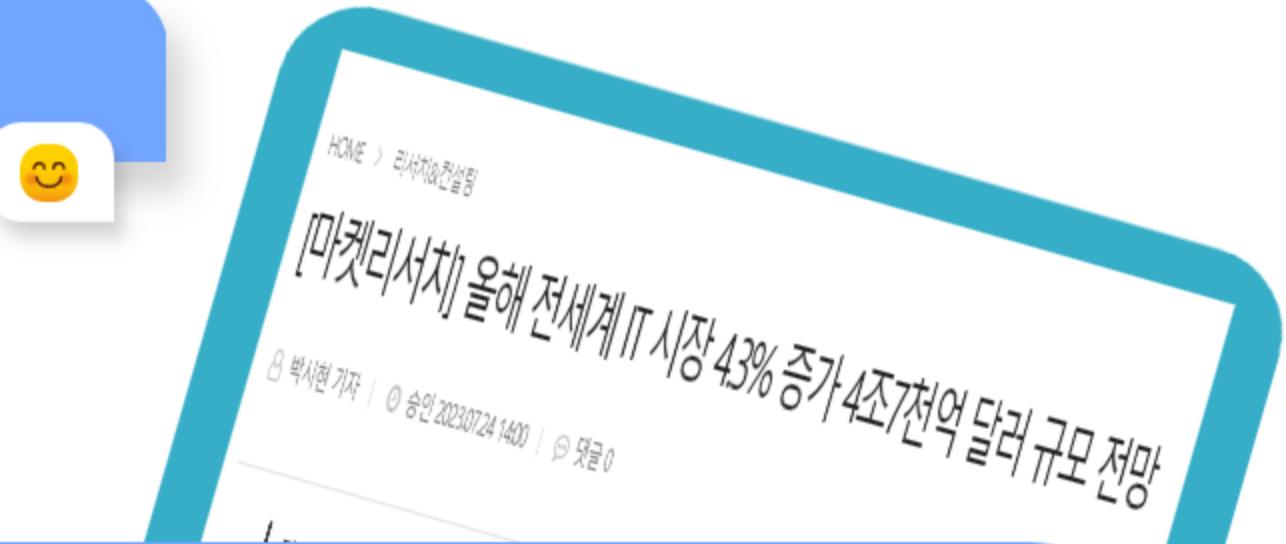
IT 시장 성장
확대 전망



(단위:십억원)



Source: IDC Quarterly Enterprise Infrastructure Tracker, June 2023





#프로젝트 개요

AI기술을 활용한 소프트웨어 엔지니어링 과정

#주제 선정 배경



- IT시장의 성장으로 인한 프로젝트 관리 시스템 수요 증가
- 복잡한 서류로 인한 휴먼오류를 줄이기 위한 솔루션
- 지속적인 IT분야의 성장으로 개발자를 위한 PMS도입



#프로젝트 개요

AI기술을 활용한 소프트웨어 엔지니어링 과정

#요구사항 #ROLE



사원

- 오늘의 일정
- 프로젝트
- 프로젝트 상세
- 일감
- 이슈
- 파일저장소
- 위키조회
- 공지사항 조회
- 서식 조회
- 커뮤니티
- 마이페이지

프로젝트 관리자

- 프로젝트 생성
- 프로젝트 관리
- 팀원초대
- 프로젝트 접근제한
- 위키등록

시스템 관리자

- 로그인(얼굴인식)
- 프로젝트 관리
- 일감 관리
- 사원 관리
- 공지사항 관리
- 서식 관리
- 게시판 관리
- Q&A 관리



#프로젝트 개요

AI기술을 활용한 소프트웨어 엔지니어링 과정

#기대 효과

14:00 pm



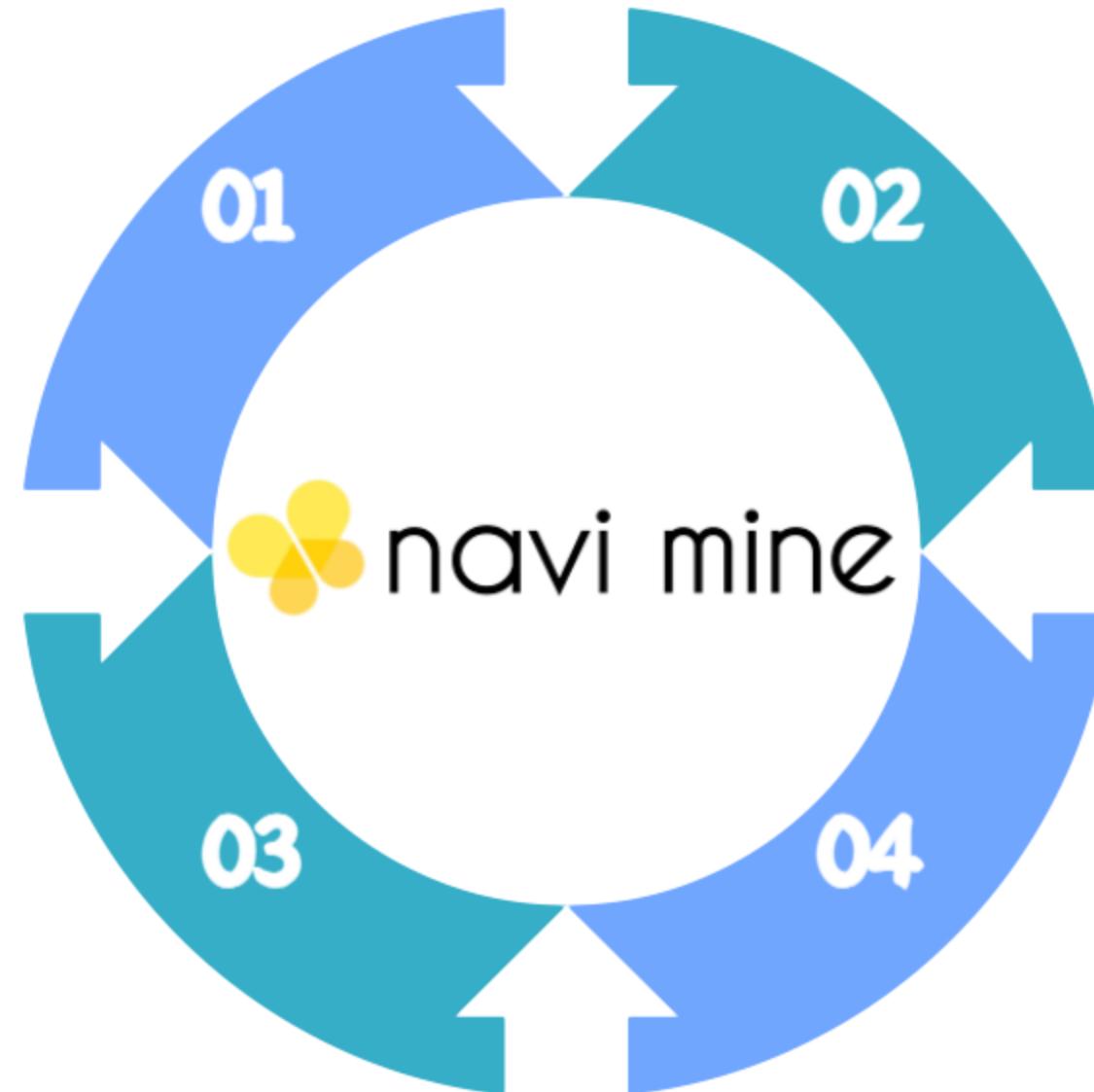
손쉬운 프로젝트 관리

오늘의 일정을 통한 손쉬운
프로젝트 관리



다양한 콘텐츠

추천 서비스 나눔, 포토게시판 등
여러 콘텐츠를 이용할 수 있음



원활한 커뮤니케이션

채팅과 알림 기능을 통한
커뮤니케이션 제공



업무 효율성 증가

업무, 소통, SNS 기능의
삼위일체 NAVI MINE





#프로젝트 개요

AI기술을 활용한 소프트웨어 엔지니어링 과정

#개발 환경



HTML



JS



CSS



ORACLE®

python™

eclipse

MyBatis

HIBERNATE



maven



03 수행 절차

T E A M 3 N A V I M I N E P M S



#수행 절차

AI기술을 활용한 소프트웨어 엔지니어링 과정

피드백 주시거나 내용 수정해서 업로드해주세요

주요일정	일정 내용	비고
팀구성	1월 2일	프로젝트 팀구성
주제선정	1월 3일	주제선정 회의진행 및 포로젝트 기획안 제출
	1월 8일	주제 확정 및 최종 기획안 제출
설계	1월 9일	설계산출물 작성 시작 및 보고
	2월 7일	프로젝트 착수 발표 진행
구현기간	2월 8일	설계 산출물에 따른 개발 로직 구현
	2월 16일	진행 보고 및 중간 테스트
테스트 및 배포기간	3월 28일	프로젝트 테스트, 최종발표 자료 준비, 리허설
	4월 1일	프로젝트 최종발표일

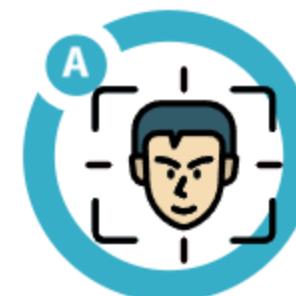


#수행 절차

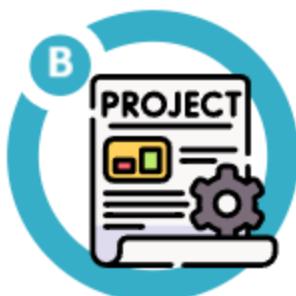
#주요 기능

AI기술을 활용한 소프트웨어 엔지니어링 과정

시스템 관리자



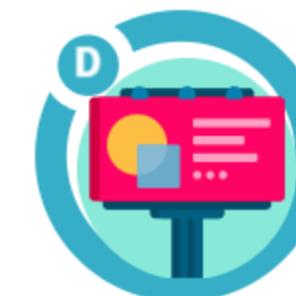
얼굴인식



프로젝트 관리

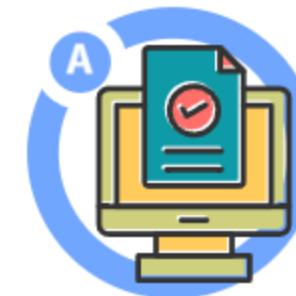


인사관리

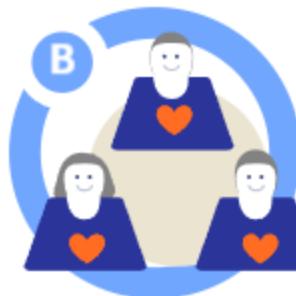


게시판 관리

프로젝트 관리자



프로젝트 등록



팀원 관리



프로젝트 관리



위키 등록

일반 사용



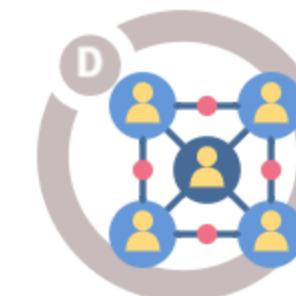
오늘의 일정



프로젝트



마이페이지



커뮤니티 Q&A

#수행 절차

#주요 기능

AI기술을 활용한 소프트웨어 엔지니어링 과정



#수행 절차

#AI 주요 기능

AI기술을 활용한 소프트웨어 엔지니어링 과정

AI 추천 시스템



음식추천



여행지추천



뉴스추천

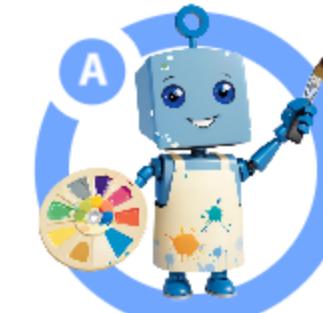


도서추천



자격증추천

AI 모듈 시스템



AI 이미지 생성



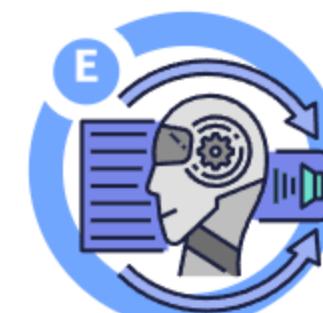
번역



얼굴인식



졸음방지



음성인식



04 산출물

T E A M 3 N A V I M I N E P M S

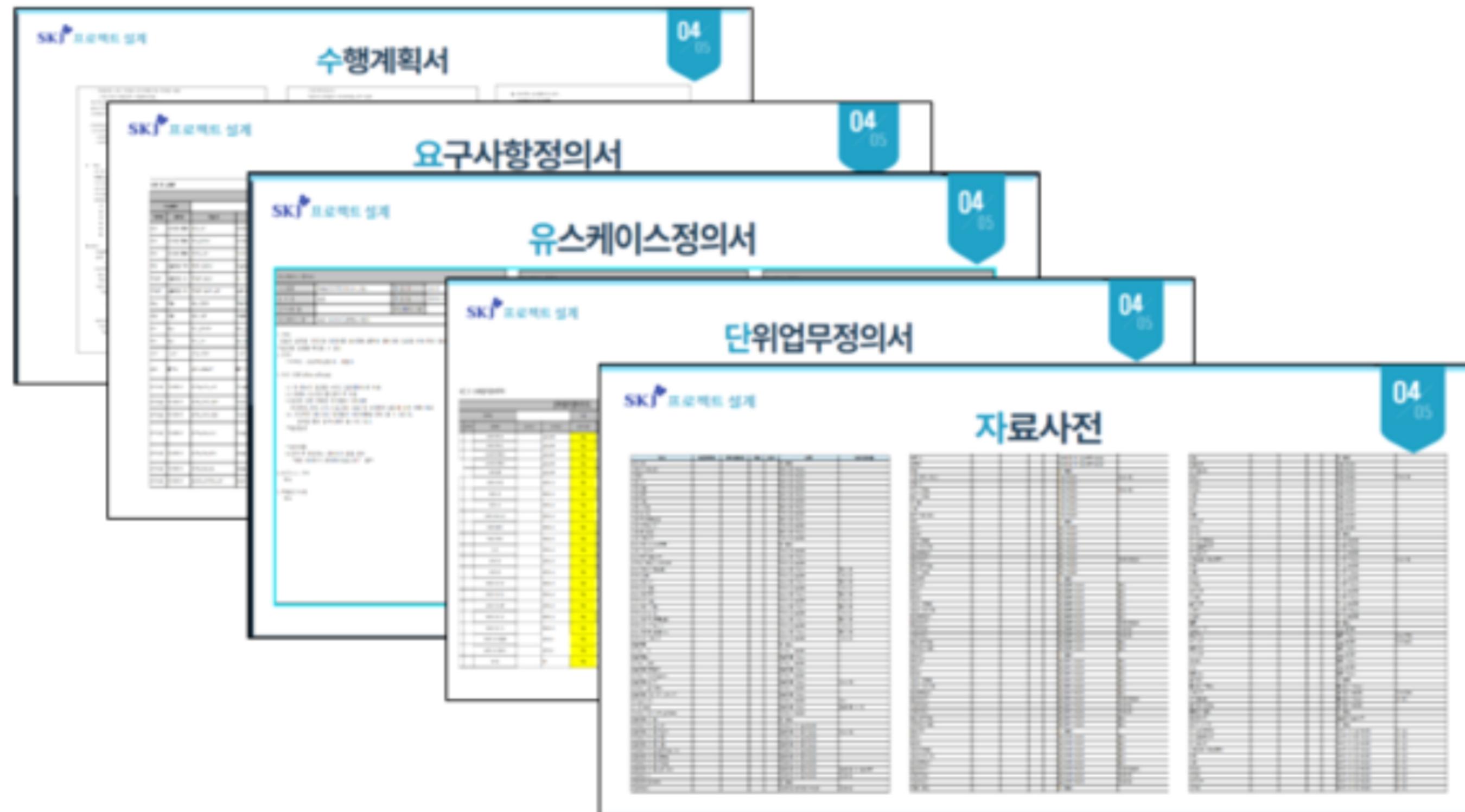


#산출물

#산출물 - 1



AI기술을 활용한 소프트웨어 엔지니어링 과정





#산출물

#산출물 - 2



AI기술을 활용한 소프트웨어 엔지니어링 과정



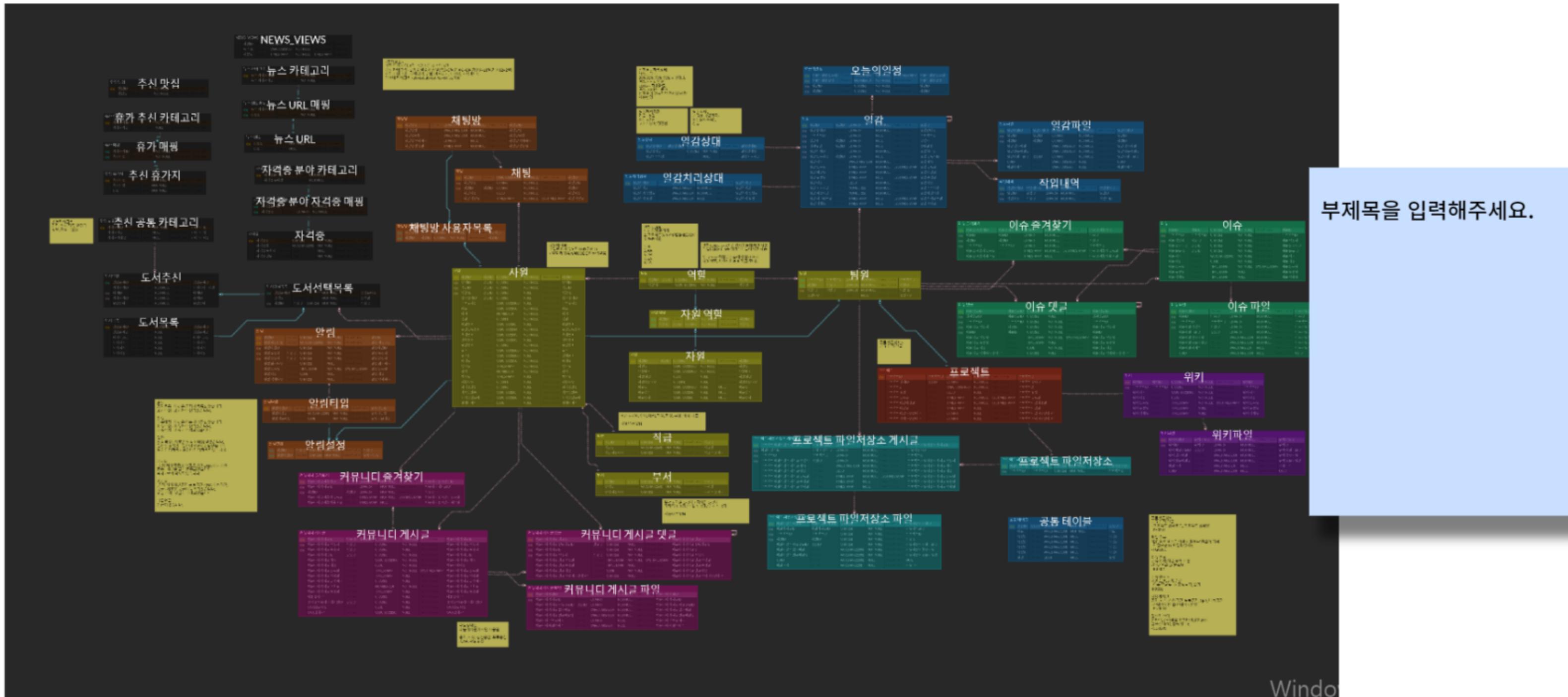


#산출물

#ERD-CLOUD



AI기술을 활용한 소프트웨어 엔지니어링 과정



Window

05 AI

T E A M 3 N A V I M I N E P M S





#AI

AI기술을 활용한 소프트웨어 엔지니어링 과정

#자격증 추천



자격증추천

자격증 명 : 전자계산기조작증용기사
자격 구분 : 국가자격

자격증설명 : 전자계산기조작증용기사는 컴퓨터 시스템의 하드웨어 구성과 시스템 운영을 위한 응용소프트웨어의 설계 및 구성을 따른 효율적인 전산 시스템을 설치, 운영하고 전자계산기 시스템을 유지, 보수하는 능력을 평가하여 부여하는 국가기술자격 중 기사 자격이다.

자격증 명 : 정보처리기사
자격 구분 : 국가자격

자격증설명 : 소프트웨어 개발 관련 자격증으로, 정보시스템의 생명주기 전반에 걸친 프로젝트 업무를 수행하는 직무로서 계획수립, 분석, 설계, 구현, 시험, 운영, 유지보수 등의 업무를 수행할 수 있는 능력을 검증하는 시험이다. 2020년부터 NCS 표준에 맞춰 시험 과목이 편기, 실기 모두를 포함해 대폭 개편되었다. 따라서 이 문서 많은 부분이 2019년 이전 정보처리기사에 대해 쓰여져 있고 협상에 공개되어 있는 요약 내용이나 기출문제 또한 2019년 이전 정보처리기사를 기준으로 하고 있으므로 정보를 검색할 때 유념해야 한다.

자격증 명 : CBT
자격 구분 : 국제자격

자격증설명 : 2017년부터 CISA/CISM/CGEIT/CRISC 시험이 CBT(Computer-Based Testing)으로 변경되었습니다. CBT라고도 알려진 이 시험은 EPSO 응시자의 언어적, 수치적, 주상적 주된 기술을 테스트하는 객관식 시험입니다. 컴퓨터 기반 시험의 단점은 응시자가 문제의 일부를 고조 표시하거나 뒤으로 표시하거나 표시할 수 없다.

1/4 [=====>.....] - ETA: 0s - loss: 1.8672 - accuracy: 0.2812
4/4 [=====] - 0s 4ms/step - loss: 1.9779 - accuracy: 0.2400
2024-02-26 14:10:48.763929: I tensorflow/core/grappler/optimizers/custom_graph_optimi

1/1 [=====] - ETA: 0s
1/1 [=====] - 0s 74ms/step
0

Model: "sequential"

Layer (type)	Output Shape	Param #
<hr/>		
flatten (Flatten)	(None, 3)	0
dense (Dense)	(None, 256)	1024
dense_1 (Dense)	(None, 256)	65792
dense_2 (Dense)	(None, 100)	25700
<hr/>		

Total params: 92,516
Trainable params: 92,516
Non-trainable params: 0



#AI

AI기술을 활용한 소프트웨어 엔지니어링 과정

#뉴스 추천



뉴스추천



우리 집에도 있는데 습식·스포츠 타올 일부 제품서 폼…

김태현 기자

2024.02.22. 오후 4:15

카카오 춘신위, 위법 행위로 손해 끼친 경영진 배상책…

이나연 기자

2024.02.20. 오전 9:37

서울 시립병원 8곳 진료시 간 연장…응급실 24시간 …

이유민 기자

2024.02.22. 오후 4:19

닫기

확인

```
1/1 [=====] - ETA: 0s - loss: 1.2780 - accuracy: 0.5217
1/1 [=====] - 0s 5ms/step - loss: 1.2780 - accuracy: 0.5217
Epoch 30/30
```

```
1/1 [=====] - ETA: 0s - loss: 1.2755 - accuracy: 0.5217
1/1 [=====] - 0s 20ms/step - loss: 1.2755 - accuracy: 0.5217
Model: "sequential"
```

Layer (type)	Output Shape	Param #
<hr/>		
flatten (Flatten)	(None, 3)	0
dense (Dense)	(None, 256)	1024
dense_1 (Dense)	(None, 512)	131584
dense_2 (Dense)	(None, 1024)	525312
dense_3 (Dense)	(None, 20)	20500
<hr/>		

Total params: 678,420

Trainable params: 678,420

Non-trainable params: 0



#AI

AI기술을 활용한 소프트웨어 엔지니어링 과정

#여행지 추천



추천 Top3



- 여행지명 사이판
- 순위 1위
- 안내 사이판은 전체 115.39 km²의 면적과 함께 서쪽의 태평양에 있는 마리아나 제도에 속하는 15개의 열대 섬이 하나로 늘어선, 북마리아나 제도의 가장 큰 섬이자 수도이다.



- 여행지명 몰디브
- 순위 2위
- 안내 몰디브 고하구 야치 몰디브로 나아시오 이

```
1 from flask import Flask,jsonify
2 from flask.globals import request
3 import requests
4 import json
5 import io
6 from PIL import Image
7 import os
8 from datetime import datetime
9 from boto.beanstalk import response
10 from flask import Flask
11 from flask_cors import CORS
12 from flask_templating import render_template
13
14
15 app = Flask(__name__)
16 app.config['JSON_AS_ASCII'] = False
17 CORS(app)
18 REST_API_KEY = '4435add9fe7c0694f63c5dc0f2d3582'
19
20 @app.route('/')
21 @app.route('/Karlo')
22 def karlo():
23     prompt = request.args.get('prompt')
24     r = requests.post(
25         'https://api.kakaobrain.com/v2/inference/karlo/t2i',
26         json = {
27             "version": "v2.1",
28             "prompt": str(prompt),
29             "negative_prompt": "",
30             "height": 1024,
31             "width": 1024
32         },
33         headers = {
34             'Authorization': f'KakaoAK {REST_API_KEY}',
35             'Content-type': 'application/json'
36         }
37     )
38     print(r)
39     # 응답 JSON 형식으로 변환
40     response = json.loads(r.content)
41
42     print(response)
43     print("상상된 이미지 url : " + response.get("images")[0].get("image"))
44     createImg = response.get("images")[0].get("image")
45     print(createImg)
46
47     # 이미지를 저장할 경로 지정
48     save_folder = "static"
49     os.makedirs(save_folder, exist_ok=True)
50
51     # 현재 시간을 기반으로 한 고유한 파일 이름 생성
52     now = datetime.now()
53     tt = now.strftime("%Y%m%d-%H%M%S")
54
55     # 이미지 다운로드 및 저장
56     for i, image_data in enumerate(response.get("images")):
57         image_url = image_data.get("image")
58         image_name = f"create_Karlo(t2i)_({i}).png"
59         image_path = os.path.join(save_folder, image_name)
60
61         with open(image_path, 'wb') as f:
62             f.write(requests.get(image_url).content)
```



#AI

AI기술을 활용한 소프트웨어 엔지니어링 과정

#도서 추천



도서추천

도서추천 Top3



닫기

관심분야, 나이, 경력을 통해 어떤 책을 읽었는지 학습

```
1/1 [=====] - ETA: 0s - loss: 0.0198 - accuracy: 1.0000  
1/1 [=====] - 0s 4ms/step - loss: 0.0198 - accuracy: 1.0000  
Epoch 77/80
```

```
1/1 [=====] - ETA: 0s - loss: 0.0190 - accuracy: 1.0000  
1/1 [=====] - 0s 4ms/step - loss: 0.0190 - accuracy: 1.0000  
Epoch 78/80
```

```
1/1 [=====] - ETA: 0s - loss: 0.0183 - accuracy: 1.0000  
1/1 [=====] - 0s 3ms/step - loss: 0.0183 - accuracy: 1.0000  
Epoch 79/80
```

```
1/1 [=====] - ETA: 0s - loss: 0.0176 - accuracy: 1.0000  
1/1 [=====] - 0s 4ms/step - loss: 0.0176 - accuracy: 1.0000  
Epoch 80/80
```

```
1/1 [=====] - ETA: 0s - loss: 0.0169 - accuracy: 1.0000  
1/1 [=====] - 0s 4ms/step - loss: 0.0169 - accuracy: 1.0000  
Model: "sequential"
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 512)	2048
dense_1 (Dense)	(None, 512)	262656
dense_2 (Dense)	(None, 50)	25650
<hr/>		

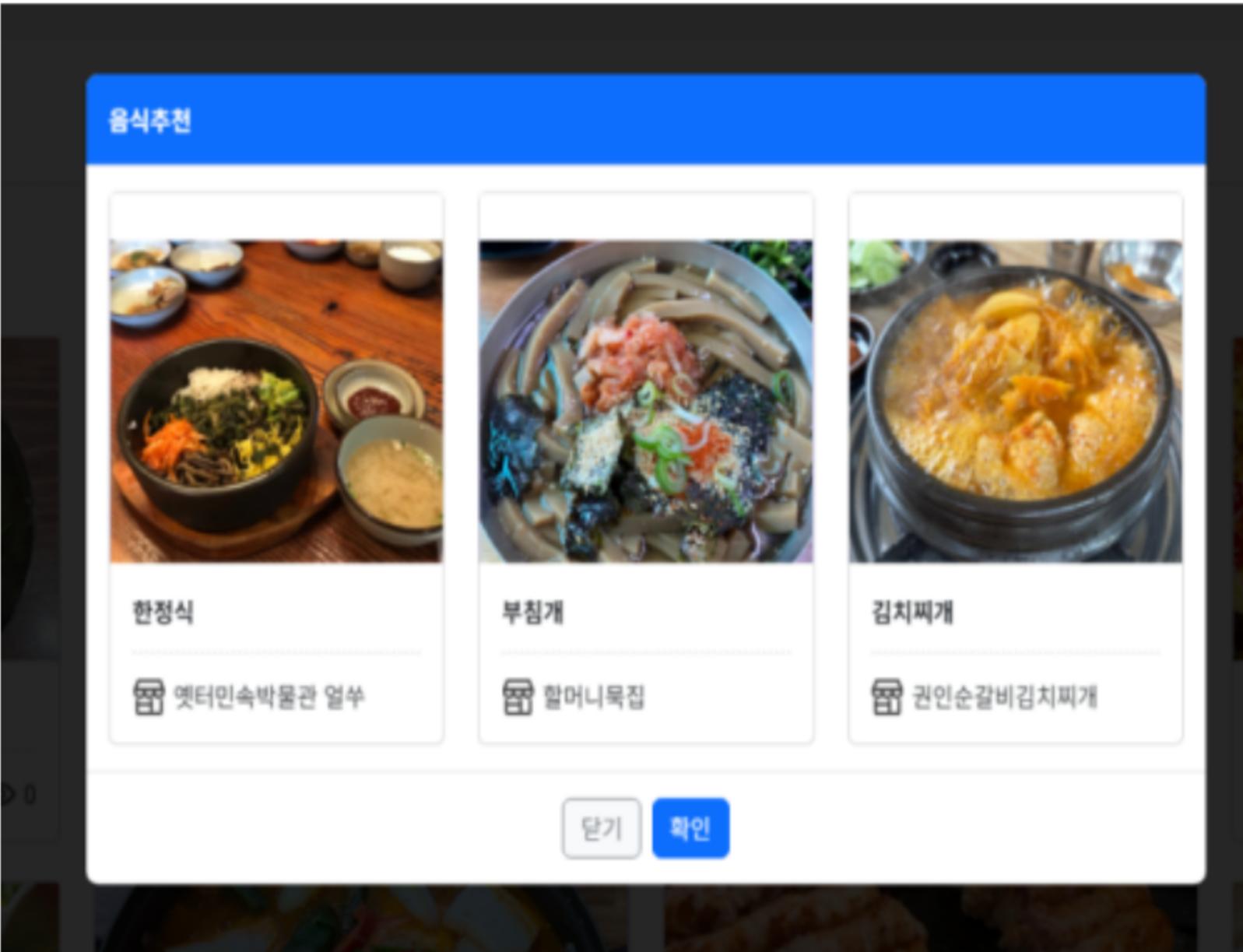
```
Total params: 290,354  
Trainable params: 290,354  
Non-trainable params: 0
```



#AI

AI기술을 활용한 소프트웨어 엔지니어링 과정

#음식 추천



```
1/8 [==>.....] - ETA: 0s - loss: 1.5975 - accuracy: 0.2812
8/8 [=====] - 0s 3ms/step - loss: 1.5081 - accuracy: 0.3568
Epoch 25/30

1/8 [==>.....] - ETA: 0s - loss: 1.6590 - accuracy: 0.3750
8/8 [=====] - 0s 3ms/step - loss: 1.5054 - accuracy: 0.3568
Epoch 26/30

1/8 [==>.....] - ETA: 0s - loss: 1.6631 - accuracy: 0.3125
8/8 [=====] - 0s 3ms/step - loss: 1.5037 - accuracy: 0.3568
Epoch 27/30

1/8 [==>.....] - ETA: 0s - loss: 1.3748 - accuracy: 0.3438
8/8 [=====] - 0s 3ms/step - loss: 1.4920 - accuracy: 0.3568
Epoch 28/30

1/8 [==>.....] - ETA: 0s - loss: 1.6345 - accuracy: 0.2812
8/8 [=====] - 0s 3ms/step - loss: 1.5001 - accuracy: 0.3568
Epoch 29/30

1/8 [==>.....] - ETA: 0s - loss: 1.6563 - accuracy: 0.2188
8/8 [=====] - 0s 3ms/step - loss: 1.5043 - accuracy: 0.3568
Epoch 30/30

1/8 [==>.....] - ETA: 0s - loss: 1.5215 - accuracy: 0.2812
8/8 [=====] - 0s 3ms/step - loss: 1.5010 - accuracy: 0.3568
Model: "sequential"

Layer (type)          Output Shape         Param #
flatten (Flatten)    (None, 3)            0
dense (Dense)         (None, 128)          512
dense_1 (Dense)       (None, 256)          33024
dense_2 (Dense)       (None, 26)           6682
=====
Total params: 40,218
Trainable params: 40,218
Non-trainable params: 0
```



#AI

AI기술을 활용한 소프트웨어 엔지니어링 과정

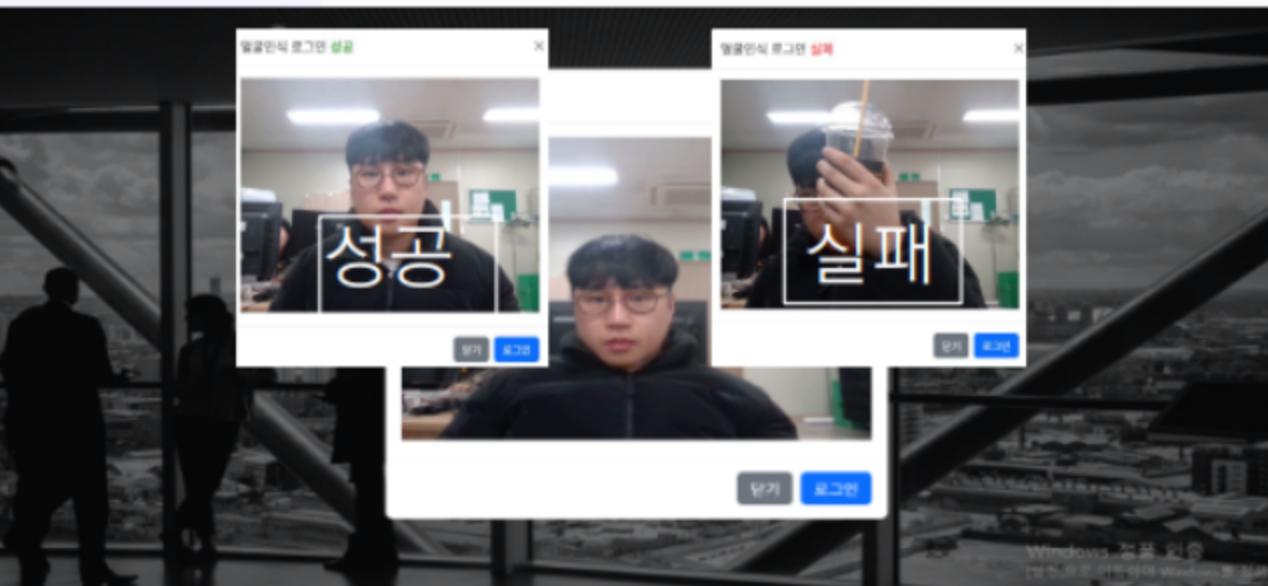
#얼굴인식



- 관리
- 사원
 - 프로젝트
 - 일감
 - 위키
 - 서식자료
 - 문자사항
 - 커뮤니티
 - 질문관리
 - 얼굴인식로그인추가
 - 얼굴인식로그인추가(사원)



사진(11) 간체업로드



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 32)	996
conv2d_1 (Conv2D)	(None, 32, 32, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_2 (Conv2D)	(None, 16, 16, 64)	18456
conv2d_3 (Conv2D)	(None, 16, 16, 64)	36920
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 64)	0
conv2d_4 (Conv2D)	(None, 8, 8, 128)	73856
conv2d_5 (Conv2D)	(None, 8, 8, 128)	147504
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 4)	8106

Total params: 295,204
Trainable params: 295,204
Non-trainable params: 0

Epoch 1/35

1/5 [=====>.....] - ETA: 0s - loss: 0.3372 - accuracy: 0.8750
5/5 [=====] - 0s 6ms/step - loss: 0.1546 - accuracy: 0.9658
Epoch 32/35

1/5 [=====>.....] - ETA: 0s - loss: 0.0950 - accuracy: 0.9688
5/5 [=====] - 0s 6ms/step - loss: 0.0680 - accuracy: 0.9932
Epoch 33/35

1/5 [=====>.....] - ETA: 0s - loss: 0.0547 - accuracy: 1.0000
5/5 [=====] - 0s 5ms/step - loss: 0.0570 - accuracy: 0.9932
Epoch 34/35

1/5 [=====>.....] - ETA: 0s - loss: 0.0190 - accuracy: 1.0000
5/5 [=====] - 0s 5ms/step - loss: 0.0458 - accuracy: 0.9932
Epoch 35/35

1/5 [=====>.....] - ETA: 0s - loss: 0.0244 - accuracy: 1.0000
5/5 [=====] - 0s 6ms/step - loss: 0.0411 - accuracy: 0.9932



#AI

#번역

AI기술을 활용한 소프트웨어 엔지니어링 과정

The figure displays three separate windows of the NAVIMINE translation application. Each window has 'NAVIMINE 번역' in the title bar and two dropdown menus for source and target languages.

- Left Window (Japanese to English):** Shows a list of Japanese lyrics from a song, with their English translations below them. Examples include "ありったけの夢をかき集め" (One piece) and "歌をかぶってた 家の地図も" (The map of the house I wore on my head).
- Middle Window (English to Korean):** Shows a list of English lyrics, with their Korean translations below them. Examples include "Ooh" (오), "You can dance" (춤출 수 있어요), and "Friday night and the lights are low" (당신 인생의 시간을 가지는 것).
- Right Window (English to English):** Shows a list of English lyrics, with their English translations below them. Examples include "Ooh" (Ooh), "You can dance" (You can dance), and "Friday night and the lights are low" (Friday night and the lights are low).

```
/*
 * 번역을 위한 컨트롤러
 * @param source 번역할 문자(text)의 나라코드
 * @param target 번역할 문자를 원하는 나라코드
 * @param text 번역할 문자
 * @return api를 통해 받은 응답
 */
@GetMapping("/trans")
@ResponseBody
public HashMap<String, Object> trans(String source, String target, String text){
    HashMap<String, Object> trans = new HashMap();
    .header("X-NCP-APIGW-API-KEY-ID", clientId)
    .header("X-NCP-APIGW-API-KEY", clientSecret)
    .header("Content-Type", "application/x-www-form-urlencoded;charset=UTF-8")
    .body("source",source)
    .body("target",target)
    .body("text",text)
    .method("POST")
    .url(apiURL)
    .build();
    return trans;
}

@Slf4j
@Controller
@RequestMapping
public class PapagoRestController {
    String clientId = "r3shn1B6hf"; // 애플리케이션_클라이언트_아이디값";
    String clientSecret = "jb9isCylH06sIe5twjIcKp1MwBdnQkqMRSk06Yd"; // 애플리케이션_클라이언트_시크릿값";
    String apiURL = "https://naveropenapi.apigw.ntruss.com/nmt/v1/translation";

    /**
     * 문자별 나라 코드를 위한 요청 응답 컨트롤러
     * @param text 사용자로 부터 받은 문자
     * @return api를 통해 받은 응답
     */
    @GetMapping("/lang")
    @ResponseBody
    public Object callAPI(@RequestParam("text") String text){
        String checkURL = "https://naveropenapi.apigw.ntruss.com/langs/v1/dect";
        HashMap<String, Object> check = new HashMap();
        .header("X-NCP-APIGW-API-KEY-ID", clientId)
        .header("X-NCP-APIGW-API-KEY", clientSecret)
        .header("Content-Type", "application/x-www-form-urlencoded;charset=UTF-8")
        .header("Accept", "application/json;charset=UTF-8")
        .method("POST")
        .url(checkURL)
        .queryString("query", text)
        .build();
        return check;
    }
}
```



#AI

AI기술을 활용한 소프트웨어 엔지니어링 과정

#AI 이미지 생성

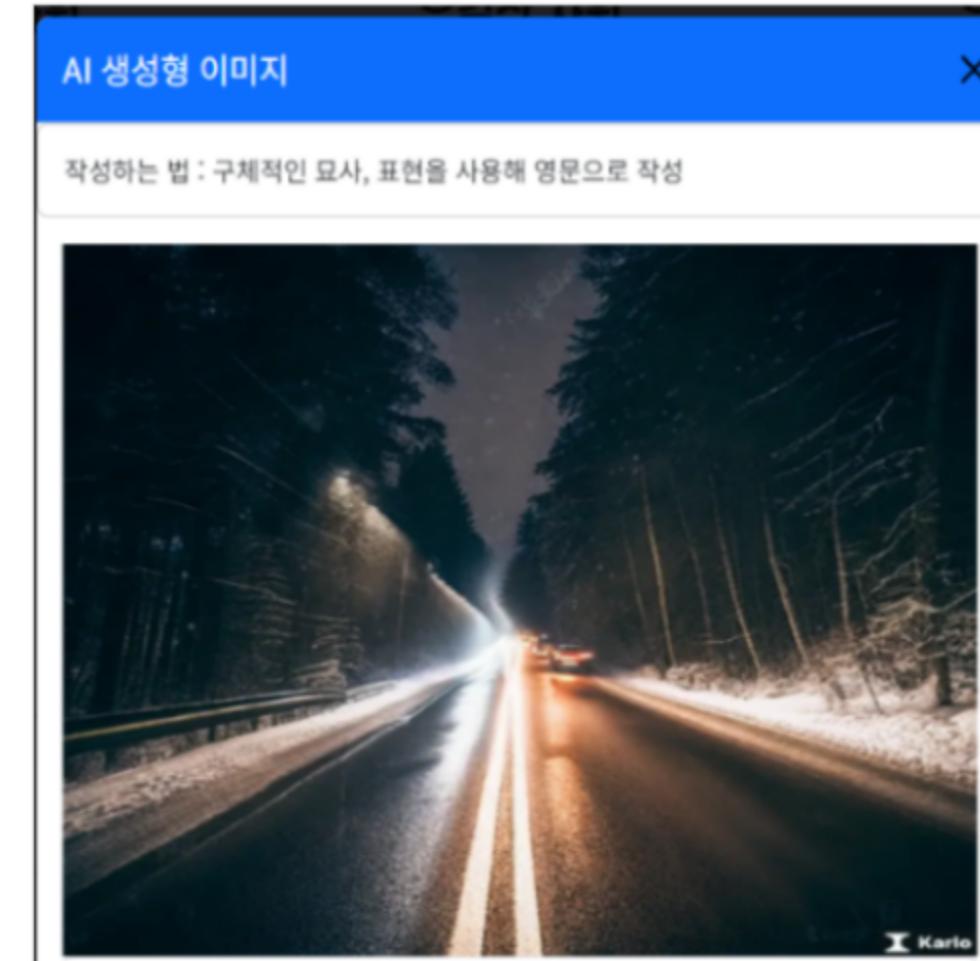


AI 생성형 이미지

작성하는 법 : 구체적인 묘사, 표현을 사용해 영문으로 작성



Big foot, Giant, Human



```
from day01.daovacation import DaoVacation  
  
model = tf.keras.models.load_model('recom.h5')  
app = Flask(__name__)  
app.config['JSON_AS_ASCII'] = False  
dv = DaoVacation()  
  
@app.route('/')  
def main():  
    age = request.args.get('age')  
    per = request.args.get('per')  
    ssn = request.args.get('ssn')  
    print('age:',age)  
    print('per:',per)  
    print('ssn:',ssn)  
    original_list=[[int(cell) for cell in row] for row in original_list]  
    recom=np.array([[int(cell) for cell in row] for row in original_list])  
    pred=model.predict(recom)  
    result=np.argmax(pred[0])  
    print(result)  
    name=dv.selectRecomName(result)  
    result_dict = {'주천여행지': name}  
    print(result_dict)  
    return jsonify(result_dict)  
  
if __name__ == '__main__':  
    app.run(debug=True)
```



#AI

AI기술을 활용한 소프트웨어 엔지니어링 과정

#졸음방지



```

for face in faces:
    print("face", face)

    # 그레이스케일 이미지에 대해 예측
    face_landmarks = predictor(gray, face)
    leftEye = []
    rightEye = []

    for n in range(36,42): # 오른쪽 눈 감자
        x = face_landmarks.part(n).x
        y = face_landmarks.part(n).y
        leftEye.append((x,y))
        next_point = n+1
        if n == 41:
            next_point = 36
        x2 = face_landmarks.part(next_point).x
        y2 = face_landmarks.part(next_point).y
        cv2.line(img_np,(x,y),(x2,y2),(0,255,0),1)

    for n in range(42,48): # 왼쪽 눈 감자
        x = face_landmarks.part(n).x
        y = face_landmarks.part(n).y
        rightEye.append((x,y))
        next_point = n+1
        if n == 47:
            next_point = 42
        x2 = face_landmarks.part(next_point).x
        y2 = face.landmarks.part(next_point).y
        cv2.line(img_np,(x,y),(x2,y2),(0,255,0),1)

    left_ear = calculate_EAR(leftEye)
    right_ear = calculate_EAR(rightEye)

    EAR = (left_ear+right_ear)/2
    EAR = round(EAR,2)

    print("테스트", EAR)

    # EAR이 일정 값보다 적으면 눈을 감은 것으로 판단
    if EAR < 0.2:
        Eye_Cnt += 1
        print("연속으로 눈감은 횟수: ", Eye_Cnt)
        # 일정 시간 동안 눈을 감은 상태면 경고
        if 5 < Eye_Cnt:
            if t == 0:
                socketio.emit('play_sound',
                {'url': 'http://192.168.141.24:5000/static/sleep.mp3'})
            t = 1
            print("!!!자고있음!!!")
    else:
        # 눈을 감은 상태가 아니면 상태를 초기화
        if Eye_Cnt != 0:
            socketio.emit('stop_sound',
            {'url': 'http://192.168.141.24:5000/static/sleep.mp3'})
        t = 0
        Eye_Cnt = 0

```

Flask

```

from flask_socketio import SocketIO
app = Flask(__name__)
CORS(app)
socketio = SocketIO(app, cors_allowed_origins="*")
@socketio.on('webcam_data')
def handle_webcam_data(image_blob):
    process_webcam_data(image_blob)

if __name__ == '__main__':
    socketio.run(app, host='192.168.141.24', port=5000)

```

IP와 port를 맞추고 webcam_data로
클라이언트의 영상이 서버쪽으로 전송됨

WebProject

```

<video id="webcam" width="640" height="480" autoplay></video>
<audio id="audio" src="" type="audio/mpeg"></audio>

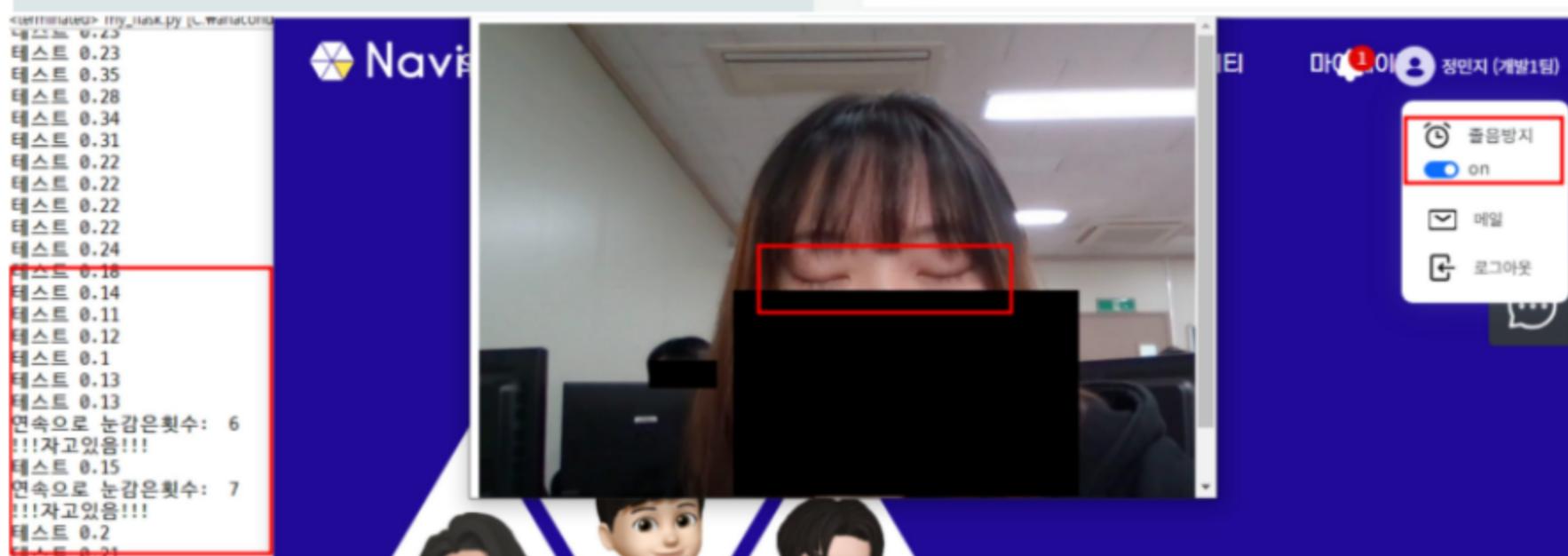
<script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.3.2/socket.io.js">
<script>
const socket = io('http://192.168.141.24:5000');

// 웹캠 스트림 가져오기
navigator.mediaDevices.getUserMedia({ video: true })
.then(function(stream) {
    var video = document.getElementById('webcam');
    video.srcObject = stream;

    var canvas = document.createElement('canvas');
    var context = canvas.getContext('2d');

    // 웹캠 영상을 실시간으로 서버로 전송
    setInterval(function() {
        context.drawImage(video, 0, 0, canvas.width, canvas.height);
        canvas.toBlob(function(blob) {
            socket.emit('webcam_data', blob);
        }, 'image/jpeg');
    }, 1000);
}).catch(function(error) {
    console.error('웹캠 접근 예리:', error);
});

```





#AI

AI기술을 활용한 소프트웨어 엔지니어링 과정

#음성인식



06 시연

T E A M 3 N A V I M I N E P M S





T E A M 3 N A V I M I N E P M S