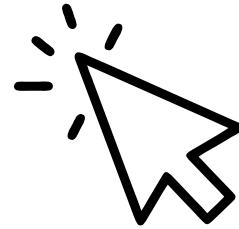


# 안전한 데이터 활용분야 특허 공모전: 산재신청 온인원 서비스, 산재메이트 데이터 활용과 개인정보 보호의 조화를 위한 앱 플랫폼



후양듀오

중앙대학교 경영학과 박서윤  
중앙대학교 심리학과 유지원



x

# 목 차

01

앱 플랫폼 소개

02

제안 배경

03

활용 데이터

04

기능 구현1: 산재 보험 지정 의료  
기관 지도 시작화

05

기능 구현2: 산재 지정의료기관  
최신 리뷰 요약 효과

06

기능 구현3: 발생확률 높은 재해유형 예측

07

앱 플랫폼 구현 예시

08

기대 효과 및 향후 계획

09

참고문헌

# 앱 플랫폼 소개

"산재 예방부터 신청, 승인, 재활까지 전 과정에서 개인정보 보호와 데이터 활용을 조화시키는 앱 플랫폼"

## 플랫폼 주요 기능

- 산재 예방 정보 제공:  
AI 기반 재해 유형 예측 및 맞춤형 정보 제공
- 산재 신청 지원:  
단계별 가이드라인 제공 및 행정 처리 자동화
- 병원 및 재활 서비스 정보 제공:  
데이터 시각화 및 최신 리뷰 요약

## 기술적 차별성

- 개인정보 보호를 강화한 데이터 분석 및 AI 활용
- 병원 및 재활 기관 데이터의 시각화  
(지도 API 연동)
- 동적 크롤링 및 생성형 AI를 통한 실시간 리뷰 요약

# 제안 배경

## 산재보험이란?

- 근로자를 업무 중 발생한 부상, 질병, 장애, 사망 등으로부터 보호하는 사회보험 제도
- 무과실책임주의로 사업주의 동의 여부와 관계없이 신청 가능

## 산재보험의 주요 문제점



### 신청 단계 문제

- 산재 은폐와 미인식으로 이용률 저조
- 신청자가 직접 처리해야 하는 불편함
- 처리 과정의 복잡성과 긴 소요 시간
- 행정 실수로 인한 문제 발생

### 치료 단계 문제

- 정보 부족으로 재활 서비스 이용 제한
- 근로복지공단 병원 및 재활 서비스 정보 미흡

### 필요성

- 데이터를 활용한 자동화 플랫폼 서비스 개발
- 산재 초기부터 요양 및 재활까지 단계별 맞춤형 정보 제공
- 앱을 통해 절차를 간소화하고 효율적 처리 지원
- 필요한 정보와 서비스를 제공하여 신속하고 정확한 치료와 재활 지원

# 활용 데이터

제공기관명	데이터명	활용 방식
근로복지공단	산재보험 지정의료기관 현황정보	<ul style="list-style-type: none"><li>• 산재 지정의료기관의 주소를 위도와 경도로 추출하여 지정의료기관 위치를 지도에 시각화(지도 시각화) 활용함</li><li>• 해당 데이터를 생성형 AI를 활용한 병원 리뷰 요약을 제시하는데 활용함으로써 산재 근로자들이 원하는 의료기관을 더욱 찾기 쉽게 함</li></ul>
근로복지공단	산재보험 직업재활기관 현황정보	산재 직업재활기관의 주소를 위도와 경도로 추출하여 지정의료기관 위치를 지도에 시각화(지도 시각화) 활용함
고용노동부	사고재해자 개별자료	성별, 연령대, 근속기간, 대업종 등의 데이터를 사용하여 발생 가능한 사고 재해를 예측하는데 활용함

# 기능구현1: 산재 보험 지정 의료 기관 지도 시각화

Out[1]:								
연도	의료기관번호	의료기관명	우편번호	주소	Latitude	Longitude	전화번호	
0	2022	1000001	연세대학교의과대학강남세브란스병원	6273	서울 강남구 연주로 211 (도곡동) 37.492549	127.046528	02-2019-2163	
1	2022	1000050	의료법인 체일병원	6664	서울 서초구 서초대로 116 (방배동) 37.487829	126.995077	02-586-5557	
2	2022	1000072	예루살렘정밀의과	6737	서울 서초구 남부순환로350길 33 (양재동) 37.463457	127.036467	02-575-0122-0	
3	2022	1000116	가톨릭장성의과의원	6109	서울 강남구 연주로 601 9층 (논현동, 파크랜드빌딩) 37.508818	127.037727	02-554-7575	
4	2022	1000125	안정현의과	6021	서울 강남구 도산대로 303 (신사동) 37.522110	127.034443	02-3444-7277	
...	...	...	...	...	...	...	...	
6054	2022	1022458	푸른길 정형외과 의원	61435	광주 동구 능명로70번길 5 (능명동) 35.150802	126.928451	062-233-9900	
6055	2022	1022469	동지과의원	54995	경북 경주시 완산구 광활로 262-6 4층 (서노승동, 세이브존)	35.825525	127.144781	053-282-7528
6056	2022	1022470	거인병원	47876	부산 동래구 어고로 12 (사직동, 거인병원)	35.194655	129.065384	051-996-7000
6057	2022	1022473	은혜요양병원	59675	전남 여수시 거북신공원2길 5-5 (학동) 34.760803	127.665965	061-688-9805	
6058	2022	1022475	삼성르동정형외과의원	13406	경기 성남시 송월구 둔촌대로 272 B1, 2~5층 (하대원동) 37.428800	127.148390	031-696-2175	



## 목적

- 사용자 편의를 위한 지정 의료기관 및 직업재활기관의 위치 정보 제공
- 재활 병원 접근성 개선

## 활용 방안

사용자들이 지도 API를 통해 인근 산재 지정 의료기관을 효율적으로 탐색 가능

## 작업 과정

- 주소 데이터 변환
  - Geopy 확장 프로그램을 활용해 주소 데이터를 위도·경도 좌표로 변환
- 지도 시각화
  - Folium API를 사용해 구글 지도에 시각화
  - 지정의료기관 및 직업재활기관의 위치를 직관적으로 표시
- 결과물 저장
  - 시각화한 지도 데이터를 HTML 파일로 저장
  - 사용자 애플리케이션에 적용하여 인근 의료기관 확인 기능 제공

# 기능구현2: 산재 지정의료기관 최신 리뷰 요약

```
: import time
import pandas as pd
import warnings
warnings.filterwarnings('ignore')

: from selenium import webdriver # 드라이버
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.keys import Keys # Enter
from selenium.webdriver.common.by import By
from bs4 import BeautifulSoup
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from selenium.common.exceptions import NoSuchElementException
```

연도 의료기관번호 의료기관명 우편번호 주소 전화번호					
0 2022 1000001 연세대학교의과대학 강남세브란스병원 6273 서울 강남구 언주로 211 (도곡동) 02-2019-2163					
1 2022 1000050 의료법인 제일병원 6664 서울 서초구 서초대로 116 (방배동) 02-586-5557					
2 2022 1000072 예루살렘정형외과 6737 서울 서초구 남부순환로350길 33 (양재동) 02-575-0122-0					
3 2022 1000116 가톨릭정형외과의원 6109 서울 강남구 언주로 601 3층 (논현동, 파크랜드빌딩) 02-554-7575					
4 2022 1000125 안성형외과 6021 서울 강남구 도산대로 303 (신사동) 02-3444-7277					
5 2022 1000147 광동한방병원 6170 서울 강남구 봉은사로 612 (삼성동) 02-2222-4883					
6 2022 1000158 한양대학교병원 4763 서울 성동구 왕십리로 222-1 (사근동, 한양대학부속병원) 02-2290-9096					
7 2022 1000159 강동성심병원 5355 서울 강동구 성안로 150 (길동) 02-2224-2127					
8 2022 1000161 국립경찰병원 5715 서울 송파구 송이로 123 (가락동) 02-3400-1160					
9 2022 1000168 건국대학교병원 5030 서울특별시 강진구 능동로 120-1 (화양동, 건국대학교병원) 02-2030-7160					

<작업과정 1.데이터 준비>

<b>목적</b>	<ul style="list-style-type: none"><li>사용자 맞춤 의료기관 리뷰 제공</li><li>산재 지정의료기관 선택을 위한 정보 접근성 향상</li></ul>
<b>활용 방안</b>	<ul style="list-style-type: none"><li>사용자 중심의 최신 리뷰 기반 의료기관 정보 제공</li><li>의료기관 선택 과정에서의 의사결정 지원</li></ul>
<b>작업 과정</b>	<p>1. 데이터 준비</p> <ul style="list-style-type: none"><li>근로복지공단 CSV 파일을 Pandas 데이터프레임(df)으로 로드</li><li>사용자가 입력한 의료기관의 인덱스를 hospital_idx 변수에 저장</li></ul>



```
# 카카오 맵으로 이동
url = "https://map.kakao.com/"
chrome_driver_path = 'C:/Users/kellykim/Downloads/chromedriver-win64/chromedriver.exe'
service = Service(chrome_driver_path)
options = Options()
options.add_argument('--headless') # 브라우저를 숨김 모드로 실행
options.add_argument('--no-sandbox')
options.add_argument('--disable-dev-shm-usage')
driver = webdriver.Chrome(service=service, options=options)
driver.get(url)

searchloc = df['의료기관명'][hospital_idx] + ' ' + ''.join(list(df['주소'][hospital_idx].split())[:2])

search_area = driver.find_element(By.XPATH, r'//*[@id="search.keyword.query"]') # 카카오맵 검색창
search_area.send_keys(searchloc) # 검색어 전달
driver.find_element(By.XPATH, r'//*[@id="search.keyword.submit"]').send_keys(Keys.ENTER) # 등보기 클릭
time.sleep(5)

driver.find_element(By.XPATH, r'//*[@id="info.main.options"]/li[2]/a').send_keys(Keys.ENTER) # 경소 버튼

registered = 1 # 카카오맵에 의료기관 등록 여부
review_exist = 1 # 후기 개수 0개 or 후기 미제공
```

## 〈작업과정 2. 웹 크롤링〉

### 작업 과정

#### 2. 웹 크롤링

- Selenium 및 ChromeDriver를 사용하여 카카오맵(map.kakao.com) 접속
- 검색창에 의료기관명과 주소를 입력하여 결과 탐색
- 의료기관 등록 상태와 리뷰 존재 여부를 판단 (registered, review\_exist)



```

# 訂H 亂H 與 ChatGPT API 亂H 訂H 亂H
def extract_review():
    html = driver.page_source
    soup = BeautifulSoup(html, 'html.parser')

    # 亂H 亂H 亂H
    review_lists = soup.select('.list_evaluation > li')

    count = 0
    rev = []

    if len(review_lists) != 0:
        for review in review_lists:
            comment = review.select('.txt_comment > span')[0].text # 亂H
            if len(comment) != 0:
                rev.append(comment)

    # 亂H 亂H 亂H 亂H
    driver.close()
    driver.switch_to.window(driver.window_handles[0])
    time.sleep(2)

    return rev

if review_exist:
    # 亂H 亂H 亂H 亂H
    driver.find_element(By.XPATH, r'//*[@id="info.search.place.list"]/li[*str(i)+"]/div[5]/div[4]/a[1]').send_keys(Keys.ENTER)
    driver.switch_to.window(driver.window_handles[-1])
    time.sleep(2)
    # 亂H 亂H 亂H 亂H
    for i in range(5):
        try:
            more_rvs = driver.find_element(By.CLASS_NAME, 'txt_more') # 亂H 亂H 亂H
            more_rvs.click()
            time.sleep(1)
        except NoSuchElementException:
            pass

    rev = extract_review() # 亂H 亂H 亂H 亂H
else:
    rev = "X"

```

```
from openai import OpenAI

# open ai에서 발급받은 api key를 등록합니다.
OPENAI_YOUR_KEY = #your open ai api key
# 사용 모델을 설정합니다. chat GPT는 gpt-3.5-turbo를 사용합니다.
MODEL = "gpt-3.5-turbo"
USER_INPUT_MSG = "다음의 하나 이상의 병원 리뷰를 요약해서 종합한 하나의 리뷰를 출력해줘. 리뷰 요약을 할 때 주체는 특정인이 아닌 제3자인 것처럼 해줘. 서술에 스타일은 .

# OpenAI API 키 설정
client = OpenAI(api_key=OPENAI_YOUR_KEY) # 사용자의 API 키로 대체해야 함
if review_exist:
    response = client.chat.completions.create(
        model = MODEL,
        messages = [{"role" : "user", "content" : USER_INPUT_MSG}]
    )
    review_summary = response.choices[0].message.content
else:
    review_summary = "X"

print(review_summary)
```

### 〈작업과정 3. 리뷰 추출 및 요약〉

## 작업 과정

### 3. 리뷰 추출 및 요약

- 의료기관 이름, 주소, 평점 추출 (예: '중앙대학교병원', '서울 동작구 흑석로 102', 평점 2.2)
  - 리뷰 더보기 버튼을 최대 5번 클릭하여 충분한 리뷰 수집
  - ChatGPT API를 활용해 리뷰 요약 → review\_summary 저장

## 4. 결과 처리

- 리뷰가 없으면 결과에 'X'를 출력
  - 최신 리뷰 요약을 사용자에게 제공

# 기능구현3:발생확률 높은 재해유형 예측

```
import pandas as pd  
  
df = pd.read_csv("/content/drive/MyDrive/고용노동부_공모전/고용노동부_사고재해자 개별자료_20221231.csv", encoding='euc-kr')  
df.head()  
  
[ 0 2022 여 30대 6개월 미만 제조업 50인~99인 사고사망자 끼임  
 1 2022 남 50대 6개월 미만 건설업 5인~9인 사고사망자 무너짐  
 2 2022 남 30대 6개월 미만 제조업 10인~19인 사고사망자 폭발파열  
 3 2022 남 60대 이상 6개월 미만 운수·창고·통신업 30인~49인 사고사망자 사업장외교통사고  
 4 2022 남 50대 1년~2년 운수·창고·통신업 10인~19인 사고사망자 사업장외교통사고  
  
[ ] # 원핫인코딩  
encoded_feature = pd.get_dummies(df)  
from sklearn.model_selection import train_test_split  
  
X_train, X_test, y_train, y_test = train_test_split(encoded_feature,target_encoded, test_size = 0.25, stratify=target_encoded)
```

〈작업과정 1. 데이터 준비 및 전처리〉

목적	<ul style="list-style-type: none"><li>개인별 발생 가능성이 높은 재해 유형 예측</li><li>사용자에게 사전 주의사항 제공</li></ul>
활용 방안	<ul style="list-style-type: none"><li>사고 예방 조치 및 주의사항 제공</li><li>산재 사고 관리 효율성 강화</li></ul>
작업 과정	<ol style="list-style-type: none"><li>데이터 준비 및 전처리<ul style="list-style-type: none"><li>'사고재해자 개별자료' CSV 파일을 Pandas 데이터프레임으로 로드</li><li>'재해유형' 라벨인코딩, 피처값 원핫인코딩</li><li>결측값 제거</li></ul></li></ol>



```
### 로지스틱 회귀
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(X_train, y_train)

from sklearn.metrics import accuracy_score
y_pred = lr.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print('로지스틱회귀 정확도:', accuracy)

### 결정트리
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(max_depth=19,min_impurity_decrease=0.0001)
clf.fit(X_train, y_train)
pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, pred)
print('결정트리 정확도:', accuracy)

### XgBoost
import xgboost as xgb
xgb_classifier = xgb.XGBClassifier(random_state=42)
xgb_classifier.fit(X_train, y_train)

y_pred = xgb_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print('XgBoost 정확도:', accuracy)
```

## 작업 과정

### 2. 모델 개발 및 평가

로지스틱 회귀: 정확도 0.85

결정 트리: 정확도 0.99 (최종 모델 선택)

XGBoost: 정확도 1.0 (비정상적 결과로 제외)

〈작업과정 2. 모델 개발 및 평가〉



```
[94] from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(encoded_feature,target_encoded, test_size = 0.2, stratify=target_encoded)

sub_input, val_input, sub_target, val_target = train_test_split(X_train, y_train, test_size=0.2, random_state=42)
print(sub_input.shape, val_input.shape)

→ (68595, 63) (17149, 63)

[96] from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier(max_depth=15)
clf.fit(sub_input, sub_target)

print(clf.score(sub_input, sub_target))
print(clf.score(val_input, val_target))

→ 0.9944893942707195
0.9936866318152652
```

### 〈작업과정 3. 과적합 여부 확인〉

```
[64] 제안된 코드에 라이선스가 적용될 수 있습니다.| 
# 하이퍼파라미터 튜닝
from sklearn.model_selection import GridSearchCV
params = {'min_impurity_decrease': [0.0001, 0.0002, 0.0003, 0.0004, 0.0005]}
gs = GridSearchCV(DecisionTreeClassifier(random_state=42), params, n_jobs=-1)
gs.fit(X_train, y_train)

dt = gs.best_estimator_
print(dt.score(X_train, y_train))
print(gs.best_params_)
print(gs.cv_results_['mean_test_score'])

→ 0.999377993406731
{'min_impurity_decrease': 0.0001}
[0.999378 0.99980096 0.99980096 0.99955216 0.99955216]

[65] 제안된 코드에 라이선스가 적용될 수 있습니다. | art1117/making-tars
best_index = np.argmax(gs.cv_results_['mean_test_score'])
print(gs.cv_results_['params'][best_index])

→ {'min_impurity_decrease': 0.0001}

[66] params = {'min_impurity_decrease':np.arange(0.0001, 0.001, 0.0001),
               'max_depth': range(5,20,1),
               'min_samples_split': range(2,100,10)}
```

### 〈작업과정 4. 최적화〉

## 작업 과정

### 3. 과적합 여부 확인

- 68,595개 훈련 세트와 17,149개 검증 세트 비교
- 미세한 차이로 과적합으로 보기 어려움

### 4. 최적화

- 그리드서치로 최적 파라미터 탐색
- 최적 파라미터: max\_depth=19, min\_impurity\_decrease=0.0001
- 점수 0.998까지 향상

● 제안된 코드에 라이선스가 적용될 수 있습니다. |  
gs = GridSearchCV(DecisionTreeClassifier(random\_state=42), params, n\_jobs=-1)  
gs.fit(X\_train, y\_train)

✓ [69] print(gs.best\_params\_)

→ {'max\_depth': 19, 'min\_impurity\_decrease': 0.0001, 'min\_samples\_split': 2}



```
# 테스트할 샘플 데이터 추출  
sample_data = encoded_feature.iloc[5,:].to_frame().T  
sample_data  
  
연도 성별 성별_연령대 연령대_10대 연령대_20대 연령대_30대 연령대_40대 연령대_50대 연령대_60대 이상 근속기간_10년 이상 ... 재해유형_사업장내교통사고 재해유형_사업장외교통사고 재해유형_신고_산소결핍 재해유형_이상온도물체접촉 재해유형_절단배임질립 재해유형_체육행사 재해유형_폭력행위 재해유형_폭발파열 재해유형_화재 재해유형_화학물질누출접촉  
5 2022 True False False False True False False ... False  
1 rows x 63 columns  
  
[59] # 예측  
y_pred = clf.predict(sample_data)
```

```
y_pred[0]  
8  
[131] target_encoded[5]  
8  
df_target.iloc[y_pred[0],0]  
'무너짐'
```

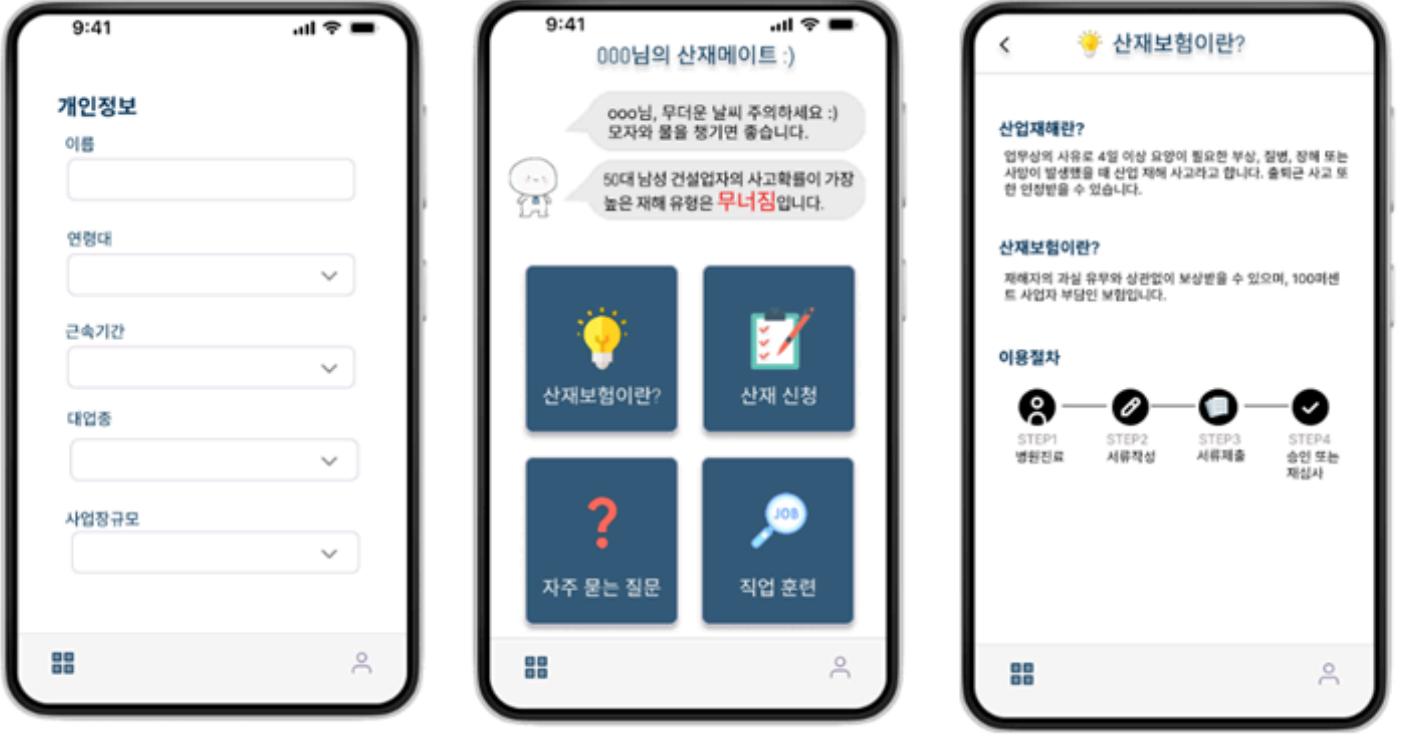
<작업과정 5. 모델 예측 결과>

## 작업 과정

### 5. 모델 예측 결과

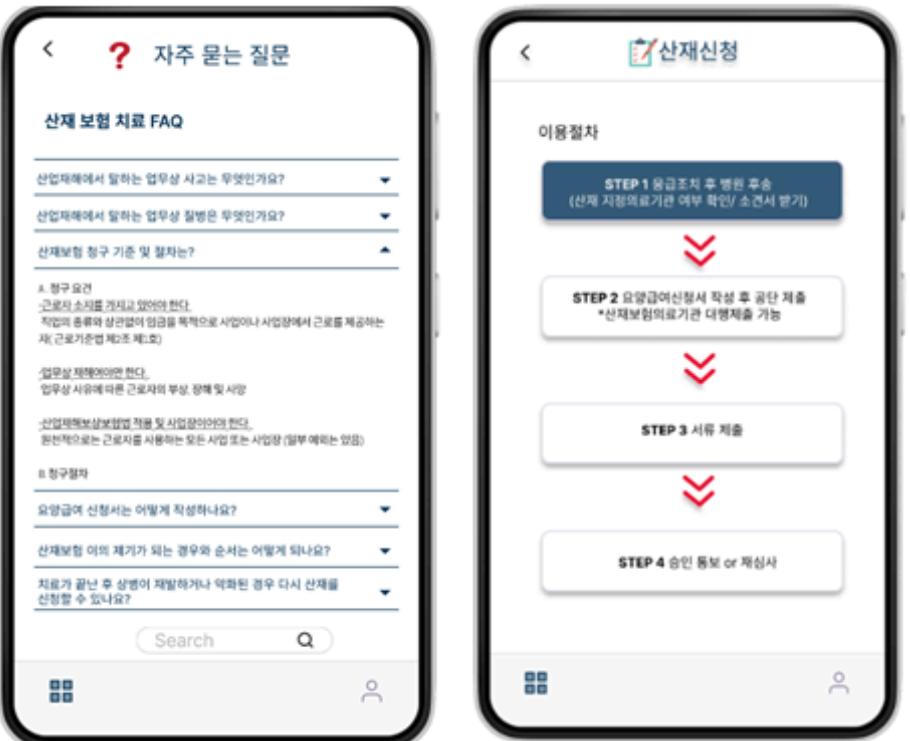
- Test 데이터의 5번째 값 예측 → ‘무너짐’ (정답과 일치)
- 107,180개 데이터 기반으로 연도, 성별, 연령대 등 다양한 변수를 활용하여 개인별 발생 가능성이 높은 재해 유형 예측

# 앱 플랫폼 구현 예시



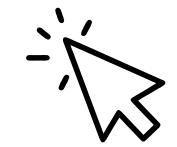
## ① 산재 예방을 위한 정보제공

- 매일 그날의 날씨를 바탕으로 주의할 점을 알림
- 입력한 개인정보(연령대, 근속 기관, 대업종 등)을 바탕으로 발생확률이 높은 재해 유형을 예측
- 현재 사용자의 상황에서 발생확률이 가장 높은 재해 유형을 알림



## ② 산재 보험에 대한 정보제공

- '산재보험이란?' 탭에서 산업재해, 산재보험, 이용 절차를 설명하여 산재보험의 개념 이해 도움
- '자주 묻는 질문' 탭을 통해 산재 처리 과정에서 발생하는 궁금증을 해소



The screenshots show the search process for a hospital. The first screen shows a map with several hospital icons. The second screen shows a list of hospitals with their names, addresses, and ratings. The third screen displays detailed information about a selected hospital, including its name, address, and a recent review.

### ③ 산재신청에 실질적 도움 제공: 4단계별 가이드라인을 제공

#### STEP1

산재 지정 의료 기관을 리스트나 지도에서 쉽게 검색하고,  
최신 리뷰 요약을 통해 선택을 도움

The screenshots show the download process of a claim form. The first screen shows the download button for the 'Claim Form.hwp' file. The second screen shows the 'Composition Method' section, which provides instructions on how to fill out the form.

요양급여신청서를 어플에서 다운로드하고 작성 방법을 참고하여

작성할 수 있도록 함

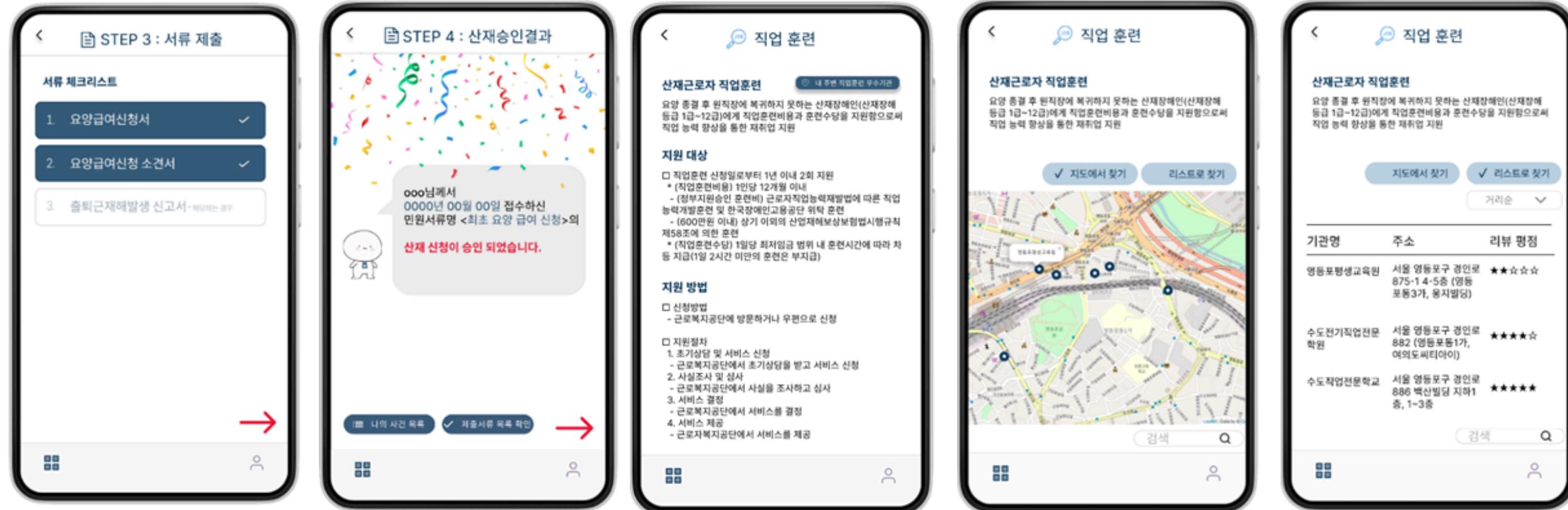
#### STEP3

제출할 서류가 모두 준비되었는지 확인 가능

#### STEP4

산재 승인 결과를 확인하고, 승인 시 직업훈련 및 재활기관 정보를 제공.

반려 시 재신청을 지원하도록 함



#### ④ 직업훈련에 대한 정보제공

- 재활과 연관된 직업 훈련 기관 및 프로그램에 대한 정보를 제공
- 근로자들이 필요한 시기에 적절한 서비스를 받을 수 있도록 함

# 기대 효과 및 향후 계획

## 사용자 중심 기대 효과

- 근로자:  
산재 예방 효과 및 편리한 행정 처리
- 사업주:  
산재 제도 이해도 향상 및 업무 효율성 증대

## 사회적 기대 효과

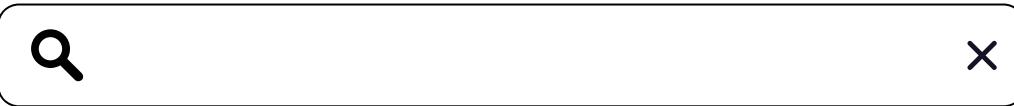
- 행정 비용 및 처리 시간 단축
- 개인정보 보호와 데이터 활용의 모범 사례 제시
- 산재 제도 및 서비스 접근성 향상

## 향후 계획

- 사용자 피드백 수집 및 개선
- 추가 데이터 연계 및 기능 확장  
(예: 맞춤형 알림 서비스)
- 서비스 상용화를 위한 단계별 추진

## 참고 문헌

- 김수현, 이정은 & 이자호.(2020) 산재근로자의 직업복귀 과정에 관한 연구: 원직장 복귀를 중심으로, *장애와 고용*, 30(1), 121–158
- 법제처. (2024.06.15). 산업재해보상보험 개관. 찾기쉬운 생활법령정보.  
<https://easylaw.go.kr/CSP/CnpClsMain.lafcsmSeq=570&ccfNo=1&cciNo=1&cnpClsNo=1>
- 한방산재케어. 산재보험치료 FAQ. 한방산재케어. <https://www.sanjaecare.com/faq>



감사합니다