

Open API 이용

장애인 전용 체육 시설 찾기

[김지원 도성탁]

CONTENTS

- 01 주제 선정
이유
- 02 제작 과정
- 03 결과 및
시연
- 04 보완할
점

주제 선정 이유

REASON 01

접근성 향상

장애인들이 체육시설을
더 쉽게 찾고
이용할 수 있도록 돕기
위해.

REASON 02

시간 절약

일일이 검색할 필요 없이
빠르게 정보를 제공해
사용자들의 시간을 절약하기
위해.

REASON 03

개인적 만족

사회적으로 유용한 도구를
직접 개발하면서
성취감과 만족감을 느끼기
위해.

제작 과정

1. Open API 이용



```
{
  "반대시설": null,
  "번호": 1,
  "소재지": "관악구",
  "시_도": "서울",
  "시설명": "SRC 서울센터",
  "운영기관": "사북) SRC",
  "장애인스포츠강좌 바우처시설": "Y",
  "전화번호": "02-871-3636",
  "홈페이지": "seoulsrc.com"
},
{
  "반대시설": null,
  "번호": 2,
  "소재지": "송파구",
  "시_도": "서울",
  "시설명": "서울곰두리체육센터",
  "운영기관": "사)한국시각장애인연합회",
  "장애인스포츠강좌 바우처시설": "Y",
  "전화번호": "02-404-6240",
  "홈페이지": "www.sgsc.or.kr"
},
```

제작 과정

2. JSON 파일로 변환

```
chatbot4.py > normalize_city_name
1  import subprocess
2  import json
3  import streamlit as st
4  import requests
5  import pandas as pd
6
7  # API URL 설정
8  base_url = 'https://api.odcloud.kr/api/15071029/v1/uddi:7c6a4eaa-179a-469e-bb19-cd39e221190c'
9  service_key = 'BQBdGk7ivmswO2MVuwxRXPForbZ%2F3vUjq1QgzyqK7exzFmS5F54XRmmrvGv6v8TXIcwT8lMVYbtkkt%2B14YOHeEDg%3D%3D'
10
11  all_data = [] # 모든 데이터를 저장할 리스트
12
13  # 페이지를 반복하여 호출
14  page = 1
15  while True:
16      url = f"{base_url}?page={page}&perPage=73&serviceKey={service_key}"
17      response = requests.get(url)
18
19      if response.status_code == 200:
20          data = response.json()
21          all_data.extend(data['data']) # 데이터를 리스트에 추가
22          if len(data['data']) < 73: # 데이터가 더 이상 없으면 종료
23              break
24          page += 1 # 다음 페이지로 이동
25      else:
26          st.error(f"API 호출 실패: {response.status_code}")
27          break
```

제작 과정

3. 시_도 이름 정규화

시_도 이름 정규화 함수 (API에 있는 시_도 기반)

```
def normalize_city_name(city_name):
```

```
    city_name = city_name.replace(" ", "") # 공백 제거
```

```
    if "서울" in city_name:
```

```
        return "서울"
```

```
    elif "부산" in city_name:
```

```
        return "부산"
```

```
    elif "대구" in city_name:
```

```
        return "대구"
```

```
    elif "인천" in city_name:
```

```
        return "인천"
```

```
    elif "광주" in city_name:
```

```
        return "광주"
```

```
    elif "대전" in city_name:
```

```
        return "대전"
```

```
    elif "울산" in city_name:
```

```
        return "울산"
```

```
    elif "세종" in city_name:
```

```
        return "세종"
```

```
    elif "경기" in city_name:
```

```
        return "경기도"
```

```
    elif "강원" in city_name:
```

```
        return "강원도"
```

```
    elif "충북" in city_name:
```

```
        return "충청북도"
```

```
    elif "충남" in city_name:
```

```
        return "충청남도"
```

```
    elif "전북" in city_name:
```

```
        return "전라북도"
```

```
    elif "전남" in city_name:
```

```
        return "전라남도"
```

```
    elif "경북" in city_name:
```

```
        return "경상북도"
```

```
    elif "경남" in city_name:
```

```
        return "경상남도"
```

```
    elif "제주" in city_name:
```

```
        return "제주도"
```

```
    else:
```

```
        return city name
```


제작 과정

4. 챗봇 인터페이스

```
# Streamlit 인터페이스
st.title('장애인 전용 체육시설 찾기 🏹')
user_input = st.text_input("찾으시려는 지역을 이야기 해주세요.(ex 서울시 성동구) :")

if user_input:
    def chatbot_with_ollama(user_input):
        # 사용자 입력에 role 추가 (user 역할로 전달)
        role_based_input = f"role: user\ncontent: Find sports facilities for disabled people in {user_input}."

        # Ollama CLI를 통해 모델을 실행하고 결과를 받아옴
        ollama_command = ["ollama", "run", "llama2", "--prompt", role_based_input]

        # subprocess로 Ollama CLI 호출
        result = subprocess.run(ollama_command, stdout=subprocess.PIPE)

        # 결과를 문자열로 변환
        llm_output = result.stdout.decode('utf-8')

        return llm_output

    # 결과 처리
    response = chatbot_with_ollama(user_input)

    # 예시: 응답으로 받은 데이터를 DataFrame에 넣을 수 있도록 변환
    results = []
```

제작 과정

4. 챗봇 인터페이스

```
# 예시: 응답으로 받은 데이터를 DataFrame에 넣을 수 있도록 변환
results = []

# '시_도'와 '소재지'를 기준으로 검색
for item in all_data:
    si_do = (item.get('시_도') or "").replace(" ", "") # 시_도 값에서 공백 제거
    so_jae_ji = (item.get('소재지') or "").replace(" ", "") # 소재지 값에서 공백 제거

    # 시_도 정규화 및 비교
    normalized_si_do = normalize_city_name(si_do)

    # 사용자 입력과 시설 정보 비교
    if normalize_city_name(user_input) in normalized_si_do or normalize_city_name(user_input) in so_jae_ji:
        results.append({
            '시설명': item['시설명'],
            '주소': f"{normalized_si_do} {so_jae_ji}",
            '전화번호': item['전화번호'],
            '홈페이지': item.get('홈페이지', '없음')
        })

if results:
    # 결과를 DataFrame으로 변환하여 테이블 출력 (인덱스를 1부터 시작)
    df = pd.DataFrame(results)
    df.index = df.index + 1
    st.table(df) # Streamlit의 테이블 형태로 출력
else:
    st.write("찾으시는 지역에는 체육 시설이 없습니다. 다른 곳에서 찾으시겠습니까?")
```


결과 및 시연

이름이 겹치는 지역도 모두
표시됨

장애인 전용 체육시설 찾기

찾으시려는 지역을 이야기 해주세요.(ex 서울시 성동구):

광주에 있는 체육 시설 정보 알려줘

	시설명	주소	전화번호	홈페이지
1	광주광역시장애인종합체육관	광주 북구	062-513-0977	www.kjwc.or.kr
2	광주광역시장애인국민체육센터	광주 서구	062-600-3900	www.gjsad.or.kr
3	성분도복지관(체육관)	경기도 광주시	031-799-0318	www.bundoreh.com
4	SRC재활체육관	경기도 광주시	031-799-3891	www.srcsportscenter.com
5	광주북구반다비체육센터	광주 북구	062-524-0731	www.gbfmtc.or.kr

보완할 점

일방적인 결과
도출

- '소재지'가 아닌 '시_도'에 따른 결과가
나타남.

장애인 전용 체육시설 찾기

찾으시려는 지역을 이야기 해주세요.(ex 서울시 성동구):

서울시 강동구 시설 알려줘

	시설명	주소	전화번호	홈페이지
1	SRC 서울센터	서울 관악구	02-871-3636	seoulsrc.com
2	서울곰두리체육센터	서울 송파구	02-404-6240	www.sgsc.or.kr
3	서부재활체육센터	서울 은평구	02-388-6622	www.sbsports.or.kr
4	동천재활체육센터	서울 노원구	02-949-9114	www.dcsports.co.kr
5	마포푸르메스포츠센터	서울 마포구	02-6070-9270	www.purmesports.or.kr
6	서울수중재활센터	서울 강동구	02-440-5700	www.seoulreports.or.kr
7	기쁜우리체육센터	서울 강서구	02-3663-8114	www.gibbunsports.or.kr
8	정립회관(체육센터)	서울 광진구	02-446-1237	www.jeongnip.or.kr

THANK YOU

감 사 합 니 다