

1. 아이디어 제목: 해양쓰레기 데이터 분석을 통한 플로깅 앱 개발

2. 활용데이터셋

파일데이터명	제공기관명	출처
2기 남해 8개정점(2018년 1차 ~ 2021년 6차).csv	연안 빅데이터 플랫폼	https://www.bigdata-coast.kr/gdsInfo/gdsInfoDetail.do?gdsCd=CT08OSN006&clear=N
2기 남해 14개정점(2022년 1차).csv	연안 빅데이터 플랫폼	https://www.bigdata-coast.kr/gdsInfo/gdsInfoDetail.do?gdsCd=CT08OSN006&clear=N
2기 남해 14개정점(2022년 2차).csv	연안 빅데이터 플랫폼	https://www.bigdata-coast.kr/gdsInfo/gdsInfoDetail.do?gdsCd=CT08OSN006&clear=N
2기 남해 14개정점(2022년 3차).csv	연안 빅데이터 플랫폼	https://www.bigdata-coast.kr/gdsInfo/gdsInfoDetail.do?gdsCd=CT08OSN006&clear=N
2기 남해 14개정점(2022년 4차).csv	연안 빅데이터 플랫폼	https://www.bigdata-coast.kr/gdsInfo/gdsInfoDetail.do?gdsCd=CT08OSN006&clear=N
2기 남해 14개정점(2022년 5차)2022년 09월 남해 환경피해 위협쓰레기 2022.csv	연안 빅데이터 플랫폼	https://www.bigdata-coast.kr/gdsInfo/gdsInfoDetail.do?gdsCd=CT08OSN006&clear=N
2기 남해 14개정점(2022년 6차) 2022년 11월 남해 환경피해 위협쓰레기 2022.csv	연안 빅데이터 플랫폼	https://www.bigdata-coast.kr/gdsInfo/gdsInfoDetail.do?gdsCd=CT08OSN006&clear=N
해안쓰레기 모니터링 정보(2018년~2020년)_지역별.csv	공공데이터포털	https://www.data.go.kr/data/15044012/fileData.do
해안쓰레기 모니터링 정보(2021)_지역별.csv	공공데이터포털	https://www.data.go.kr/data/15044012/fileData.do
해양환경공단_해양쓰레기 모니터링 정보_20211231.csv	공공데이터포털	https://www.data.go.kr/data/15044012/fileData.do
해양환경공단_해양쓰레기 모니터링 정보_20221231.csv	공공데이터포털	https://www.data.go.kr/data/15044012/fileData.do

3.배경 및 필요성

해양 환경 문제에 대한 심각성이 심해지는 요즘, 미세플라스틱과 함께 해양쓰레기가 생태계 위협의 주 원인으로 지적되며 새로운 환경 현안으로 떠오르고 있다. 전국 해양쓰레기의 추정량을 알아보기 위해 해양수산부와 해양환경관리공단이 공동으로 실시한 ‘전국 연안 해안쓰레기 일제조사’ 결과에 따르면 우리나라 해안 3,770km에 존재하는 해양쓰레기의 총량은 약 4100만 리터로 추정되며, 전체 쓰레기의 60%가 조사대상 해안 길이 10% 안에 모여 있는 것으로 나타났다. 또 유엔 환경 계획의 자료에 따르면 연간 100만 마리 이상의 바닷새가 폐그물 등 해양쓰레기로 폐사하거나 생존에 악영향을 받고 있다.

해양쓰레기가 미치는 문제점

-선박사고 및 조업 문제

바다에 버려진 밧줄 어망이 선박의 추진기에 감기거나, 비닐봉지가 냉각수 파이프에 빨려 들어가면 엔진에 부하가 걸려 운항을 할 수 없게 된다. 해수부에 따르면 우리나라의 선박사고 원인 중 1/10은 해양 폐기물이다. 또한 해양쓰레기 증가로 인해 조업 시 어망에 어획물만이 아니라 쓰레기까지 걸려 올라오는 경우가 자주 발생하는데, 이때 쓰레기를 일일이 골라내야 하는 번거로움이 생기며 건져올리는 과정에서 어망이 망가지는 경우도 빈번하다. 잡은 바다생물 가공 시 작은 쓰레기 조각 하나라도 들어가지 않게 하려면 더 많은 작업이 필요하고, 해양폐기물 때문에 어획량 자체가 줄어들기도 한다.

-바다생물 문제

해양폐기물은 바다생물들에게 큰 위협을 준다. 그 예로 유령 어업이 있다. 유엔 환경협회(UNEP)에 의하면 매년 바닷새가 100만 마리, 고래나 바다표범, 바다소 등 보호해야 할 해양 포유동물들이 10만 마리나 해양폐기물에 걸려 죽어간다. 또한 해양생물들이 쓰레기를 섭취하였을 경우 쓰레기가 위장에 쌓여 포만감을 주기 때문에 생물들이 먹을 것을 섭취하지 않게 돼 서서히 죽어가게 만들기도 한다.

-관광자원의 질 감소

해양쓰레기 수거를 자주 하더라도 바다로 밀려드는 쓰레기와 관광객들이 부주의하게 버리고 가는 쓰레기 때문에 결과적으로 바닷가 관광지의 경관이 훼손된다. 아무리 훌륭한 관광자원이 있더라도 깨끗한 바다환경이 뒷받침 되지 않으면 자원의 가치와 질이 떨어진다. 이미 바다로 들어간 쓰레기를 수거하고 처리하기 위해서는 육지보다 몇 배의 돈이 들어간다. 잠수부를 고용하거나 중장비를 이용하여 수거해야하고, 처리에 앞서 쓰레기에 달라붙은 담치, 굴, 따개비, 멸치 등을 제거하고 썩 바닷물도 씻어내야 한다. 이러한 이유로 재활용이나 소각, 매립에 어려움이 많다.

김경신 한국해양수산개발원 연구원의 말에 따르면 해양쓰레기의 심각성은 발생량의 많고 적음이 아니라 발생된 이후 쓰레기가 미치는 영향의 범위와 대상이 점점 확대되고 있다는 것에 있다. 예를 들어 페트병, 부표 등은 해양에서 적기에 수거되지 않는다면 작은 조각으로 파편화되어 미세플라스틱으로 변질된 후 해양 생태와 식품안전 더 나아가 사람의 건강에도 영향을 미치게 된다. 이를 방지하기 위해서는 쓰레기의

조기수거가 중요하나, ‘해양’이라는 환경의 특성상 쓰레기 위치 파악이 쉽지 않아 수거가 어려운 점이 문제 해결의 걸림돌이 되고 있다.

해양쓰레기 문제의 해결을 위한 정부와 지자체, 더 나아가 민간 기업, 시민 사회, 개인의 노력 또한 보태져야 한다. 이 중 시민 사회 및 개인이 해양쓰레기 문제 해결에 기여할 수 있는 방법 중 하나가 바로 플로깅이다.

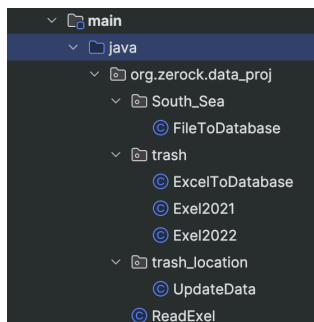
플로깅은 개인 및 시민 사회가 해양쓰레기 감소를 위해 참여할 수 있는 대표적인 운동이다. 플로깅이란 줍다라는 뜻의 스웨덴어 “**plocka upp**”와 영어 단어 조깅 “**jogging**”의 합성어이다. 단순히 쓰레기를 줍는 행위가 아닌 조깅을 하며 쓰레기를 줍는 활동인 플로깅으로 사용자들의 건강을 증진시킬 수 있다. 플로깅은 유산소, 근력 운동에 포함된다. 유산소운동 측면에서는 평균적으로 1분당 약 10~11kcal를 소모하는 것으로 밝혀졌다. 근력 운동 측면에서는 쓰레기를 줍는 동작 때문에 일반적인 로드 러닝보다 더 많은 근육을 사용한다. 특히 앉았다 일어나는 동작의 반복은 스쿼트나 런지와 비슷해 근 성장에 도움이 된다. 목포에는 현재 플로깅 센터가 있고, 한강에서도 플로깅 행사를 하곤 하지만 몇 군데로 한정되어 있으며 참여할 수 있는 방법이 다 달라 접근성이 좋지 못하다. 따라서 이번 해커톤에서 지역에 상관없이 플로깅 운동을 할 수 있게 하는 플로깅 앱을 만들 예정이다.

4. 분석내용

Java Code Explanation

- 데이터베이스 연결 코드

JDBC(Java Database Connectivity)를 사용하여 데이터베이스에 연결하는 코드를 구현했다. DriverManager 클래스를 이용하여 데이터베이스 드라이버를 로드하고, Connection 객체를 생성하여 데이터베이스에 연결했다.



- 데이터 쿼리 및 조작 코드

PreparedStatement를 활용하여 SQL 쿼리를 실행하고, ResultSet 객체를 통해 insert 과정을 수행했다. 데이터의 삽입, 갱신, 삭제와 같은 기본적인 조작에 대한 JAVA 코드 사용은 private static void retrieveData(Connection connection) 함수를 통해, console창에 해당 데이터베이스를 불러와 해결하는 방식으로 구현했다. 혹은 간단한 조작의 경우 console창에 직접 데이터들을 정리했다.

```
public class FileToDatabase {  
  
    public static void main(String[] args) {  
        String excelFolder = "/Users/admin/Desktop/공모전_데이터/South_Sea"; // 폴더 경로  
        try {  
            // Establish database connection  
            Connection connection = getConnection();  
  
            // Process all Excel files in the folder  
            processAllExcelFiles(excelFolder, connection);  
  
            // Close database connection  
            connection.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

- 에러 핸들링 및 예외 처리

유용한 데이터들을 수집하고, 이들을 데이터베이스로 옮기는 과정에서 큰 문제는 없었지만, 여러 유용한 데이터들끼리 정보를 공유하고 합치는 과정에서 문제가 발생했다. ‘지역명’ 칼럼에 대해 문제가 있었다. 어떤 데이터 베이스의 지역명이 ‘강화여차리갯벌’인 반면 다른 데이터 파일에서는 ‘강화여차리’로 되어있어 명칭이 파일마다 조금씩 다르게 정의되어있던 것이 문제였다. 이를 해결하기 위해, 아래 코드와 같이 지역명이 앞 두글자가 같으면 해당 지역에 해당하는 정보를 넣을 수 있도록 코드를 수정했다.

```
// Extract the first two characters from the region  
String regionPrefix = region.substring(0, 2);
```

데이터를 입력하고 불러오는 과정에서 앞 두 글자만을 비교하는 것은 매우 효율적인 방법이었지만, 해당 지역에 맞는 위도 경도를 불러오는 과정에서 다시한번 문제가 발생했다. ‘제주김녕리해안’, ‘제주사계리해안’ 이 두 지역이 다시 문제가 되었는데, 앞 세글자가 같으면 입력하도록 코드를 구현하였지만, 해결이 되지 않았고 이유는 다른 데이터베이스에서는 ‘제주 김녕리’ 이런식으로 띄어쓰기가 포함되어 있었기 때문이다. 그래서 모든 지역에서 띄어쓰기를 없애는 sql문을 작성하고, 지역간의 이름 유사성을 더 높이고 예외가 발생하지 않도록 지역 이름이 부분적으로 일치하는 경우에 데이터를 업데이트하는 방법으로 진행했다.

```
UPDATE South_Sea
SET invs_area_nm = REPLACE(invs_area_nm, ' ', '');
```

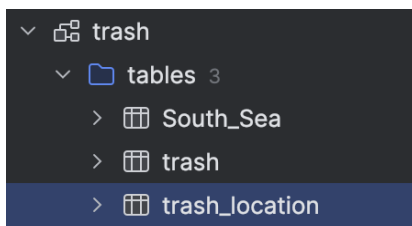
(띄어쓰기 삭제)

```
private static void updateData(Connection connection) throws SQLException {
    String selectQuery =
        "SELECT trash_location.지역, South_Sea.str_la, South_Sea.str_lo, South_Sea.end_la, " +
        "South_Sea.end_lo " +
        "FROM trash_location " +
        "JOIN South_Sea ON trash_location.지역 LIKE CONCAT('%', South_Sea.invs_area_nm, '%)";

    String updateQuery = "UPDATE trash_location " +
        "SET str_la = ?, str_lo = ?, end_la = ?, end_lo = ? " +
        "WHERE 지역 = ?";
```

Process Description

- 데이터베이스 모델 선택 및 설계



크게 trash(해양쓰레기) DataBase를 제작한 후 안에 3개의 table을 제작.

1. South_Sea는 남해 해양쓰레기와 관련된 정보로, 2022년에 6차례에 걸쳐 진행된 데이터 수집에 관한 내용이 담겨 있다.
2. Trash 테이블은 2018년부터 2022년까지의 전국 해양쓰레기에 대한 개수와 무게가 지역별로 담겨져 있다.
3. Trash_location 테이블은 trash 테이블에 프로젝트의 추가적인 활동을 위해 지역, 연도별 Count 및 Weight, 좌표 정보를 포함하여 지역기반의 통계 및 위치 관리에 효율적으로 활용될 수 있도록 제작되었다. 현재 남해의 위도, 경도만 추가된 상태이며, 이런 방식으로 전국적으로 데이터를 표시할 예정이다.

5. 해결방안

플로깅 앱 만들기

현재 플로깅 센터는 목포에 있는 보해소주 플로깅 센터 하나뿐이다. 또한 한강에서도 플로깅을 진행하긴 하지만 활성화가 잘 안 되어있는 상태이다. 플로깅에 참여하기 위해서는 여러 사이트들을 돌아다녀야 하거나 센터를 방문해야만 한다. 따라서 사람들이 지역이나 시간에 영향을 받지 않고 스스로 플로깅에 참여할 수 있도록 편리성을 제공하는 앱을 만드는 것이 목적이다. 해당 앱에서는 대한민국 해안의 해양쓰레기 양을 원 다이어그램을 통해 시각적으로 보여주고 각 지역의 플로깅 코스를 추천해준다. 또한 사용자가 제공되는 코스가 아닌 자유 코스로 플로깅을 진행할 경우 타 러닝앱과 유사하게 플로깅을 한 코스, 시간 등을 기록한다. 후에 사용자의 의사에 따라 코스를 공유할 수 있도록 만들어 다른 사용자들에게 코스를 추천하게끔 한다. 함께 플로깅 할 사람들을 모을 수 있는 온라인 커뮤니티를 만들어 플로깅을 활성화 시킬 수 있다.

#배경이 되는 대한민국 지도 plot

대한민국 시군구 shp를 다운로드 후 geojson파일로 변환했다. geopandas와 folium 라이브러리를 이용하여 대한민국 시군구 지도를 만들었다. 중심좌표를 설정한 후 folium 지도 객체를 생성했다. geopandas를 이용하여 경계를 지도에 표시하였다.

```
geojson_file = '../DataBase/TL_SCCO_SIG.json'
gdf = gpd.read_file(geojson_file)
```

```
center_lat, center_lon = 36.5, 127.5
```

```
map_korea = folium.Map(location=[center_lat, center_lon], zoom_start=7)
```

```
folium.GeoJson(gdf).add_to(map_korea)
```

#해양쓰레기의 개수 및 무게에 대한 원 다이어그램 추가

trash_location.csv 파일에는 연도별 개수, 무게에 대한 내용이 들어있으므로 연도를 설정해주고 개수 혹은 무게 중 어떤 데이터를 다이어그램으로 나타낼지 정하기 위하여 main_data를 설정해주었다. trash_location.csv 파일에 있는 모든 지역의 위도, 경도가 들어가있지 않기 때문에 시작 위도가 NULL값인 경우는 제외하였다. 원의 중심을 잡기 위하여 시작과 끝의 위도 경도의 중점을 구했다. 원의 크기를 정할때 개수의 값이 무게에 비해 현저히 높았기에 가중치를 설정해주었다.

```
Data = '../DataBase/trash_location.csv'
df = pd.read_csv(Data)
SCM = ['지역', main_data, 'str_la', 'str_lo', 'end_la', 'end_lo']
df = df[df['str_la'].notnull()] #위도, 경도 없는 지역 거르기
df = df.loc[:, SCM]

for col, row in df.iterrows():
    circle_lat, circle_lon = (row['str_la']+row['end_la'])/2, (row['str_lo']+row['end_lo'])/2
    circle_name = row['지역']
    circle_popup = f'<strong>{circle_name}</strong>'
    if count_or_weight == 'count':
        circle_radius = row[main_data]*2
    else:
        circle_radius = row[main_data]*20
    folium.Circle(location=[circle_lat, circle_lon], radius=circle_radius, popup=circle_popup,
        color='red', fill=True, fill_color='red').add_to(map_korea)
```

해양쓰레기 수거 코스 생성

1.TMap API를 이용해 보행자 도보 경로 생성 및 불러오기

```

url = "https://apis.openapi.sk.com/tmap/routes/pedestrian?version=1&callback=function"

payload = {
    "startX": startX,
    "startY": startY,
    "endX": endX,
    "endY": endY,
    "reqCoordType": "WGS84GEO",
    "startName": "%EC%B6%9C%EB%B0%9C",
    "endName": "%EB%8F%84%EC%B0%A9",
    "searchOption": "0",
    "resCoordType": "WGS84GEO",
    "sort": "index"
}

headers = {
    "accept": "application/json",
    "content-type": "application/json",
    "appKey": "e8wHh2tya84M88aReEpXC5XTQf3xgo01aZG39k5"
}

response = requests.post(url, json=payload, headers=headers)
contents = response.text
json_ob = json.loads(contents)

if 'error' in list(json_ob.keys()): # 예외 처리
    continue

```

2. 보행자 도보 경로에 해당 경로에 관한 정보를 담은 **PopUp** 생성

도보 경로를 클릭했을 때 해당 지역의 연도별 쓰레기 수거량 통계를 **Popup**하도록 전처리하는 과정이다. 추후에 해당 지역의 특성(e.g. 기후, 유동량, 수거량 등)을 종합적으로 분석해 의미 있는 **PopUp**을 추가할 계획이다.


```

# PopUp 예시 : 지역마다 연도별 쓰레기 개수 그래프 생성 (코스에 popup 형식으로 추가)

loc_name = row['지역']
year_counts = row.filter(like='count')
year_weights = row.filter(like='weight')

year_counts.index = [name[5:9] for name in list(year_counts.index)]
year_weights.index = [name[5:9] for name in list(year_weights.index)]

x = [int(year) for year in list(year_counts.index)]
y = list(year_counts.values)

fig, ax = plt.subplots()

ax.plot(x,y)
ax.set_xlabel("year")
ax.set_ylabel("trash_counts")
ax.set_title('yearly trash counts')

mpld3.save_html(fig, 'fig.html')
popup_html = open('fig.html', 'r').read()

from folium import IFrame

popup = folium.Popup(IFrame(popup_html, width=600, height=600), max_width=700)

```

3. 대한민국 지도에 추가

코스과 Popup을 대한민국 지도에 추가하는 과정이다.

```

# 보행자 도보 대한민국 지도에 추가

if geom_type == 'FeatureCollection':
    data = json_ob['features']

    for i in range(len(data)):
        raw = data[i]['geometry']
        coordinates = folium_transform(raw['type'], raw['coordinates'])
        if raw['type'] == 'Point':
            folium.Marker(location=coordinates, popup=loc_name).add_to(map_korea)
        elif raw['type'] == 'LineString':
            folium.PolyLine(locations=coordinates, popup=popup, color="orange").add_to(map_korea)
            popup = None

    elif geom_type == 'Point':
        coordinates = folium_transform(geom_type, json_ob['coordinates'])
        folium.Marker(coordinates, popup=loc_name).add_to(map_korea)
    elif geom_type == 'LineString':
        coordinates = folium_transform(geom_type, json_ob['coordinates'])
        folium.PolyLine(locations=coordinates, popup=popup, color="orange").add_to(map_korea)

# 4. HTML 형식으로 결과 저장

map_korea.save("map_path.html")
webbrowser.open("map_path.html")

```

6. 기대효과

사용자들은 힘들게 다른 사이트들을 돌아다니며 플로깅에 참여하기 위하여 노력하지 않고 간단하게 앱을 이용하여 지역이나 시간에 구애받지 않고 플로깅을 할 수 있게 된다. 커뮤니티를 통해 함께 플로깅 할 사람들을 모집하여 해양쓰레기 수거를 활성화 시킬 수 있다. 플로깅을 진행하며 각 해안 풍경의 아름다움과 해양쓰레기의 심각성을 몸소 느낄 수 있게 된다. 해당 앱을 이용함으로써 플로깅 문화를 확산시켜 해양쓰레기에 대한 관심도를 높이고 사회적 분위기를 형성할 수 있다.

7. 개선할 점

현재는 남해 해양쓰레기 데이터만 갖고 진행했지만 추후 대한민국 전체 해양쓰레기 데이터를 사용할 예정이다.

해양쓰레기의 양으로만 분석하여 진행하는 것이 아닌 각 지역별 기후, 지리정보, 쓰레기 수거량 등의 다양한 정도를 종합적으로 분석하여 가장 효율적인 코스를 도출하는 것이 보다 실용적일 것이다.

더 많은 데이터를 분석해 보다 유의미한 통계량 및 분석 데이터를 도출해 기능을 향상시킨다.

ex) 유형별(플라스틱, 끈, ..)을 지역, 연도와 연관지어 유의미한 통계량 추출, 이를 통해 보다 나은 플로깅 코스 제공

8. 참고문헌

차인환 외 4명, 플라스틱류 해양쓰레기와 생물피해 특성, 한국해양환경에너지학회, 2015

박수진, 우리나라 해양쓰레기 관리기반 현황 및 향후 정책개선 방향, 해양환경안전학회, 2014

9. GITHUB Address

<https://github.com/BDHackathon/proposal>

-> 해당 github 주소에 모든 코드 및 사용 데이터 첨부했습니다.