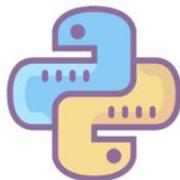


트위터 크롤링을 통한 백신별 감성분석 & FAST COVID-19 LIVE



# 데이터 수집

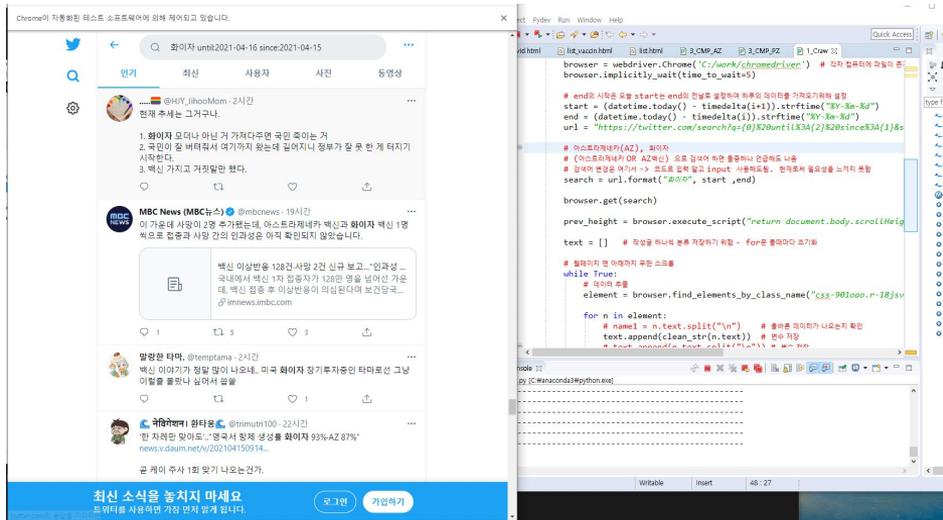


Seleniun 라이브러리를 사용해 트위터 페이지를 크롤링하여 백신별(영어/한글) 따로 수집

수집한 데이터를 가공해 감성분석



# 데이터 수집



Selenium을 이용한 백신별 Twit수집





# 데이터 가공

negative\_words\_twit - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

혈전증  
혈전  
사망  
부작용  
불가피  
더럽고  
사망자  
사퇴  
차질  
개소리  
불신  
개짓거리  
개탄  
고통  
이상반응자  
헛소리  
대유행  
근육통  
오보  
미열  
가짜  
위험  
근육통  
논란  
여지려고

positive\_words\_twit - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

호응  
돌파  
좋은  
효과  
확보  
성과  
예방  
면역  
안전한  
자궁경부암  
혈장공여  
건강한  
백신확보  
신속  
개발  
안정적  
대박  
안아픔  
대박이네  
접종  
면역  
최고  
자유  
좋다  
고피

긍정, 부정, 중립 감성별 키워드로 데이터 재가공



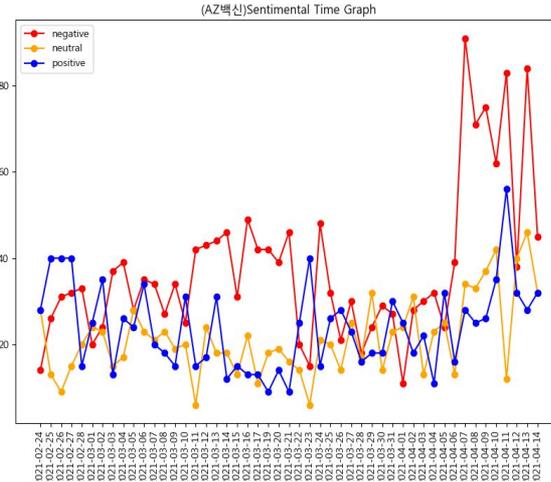
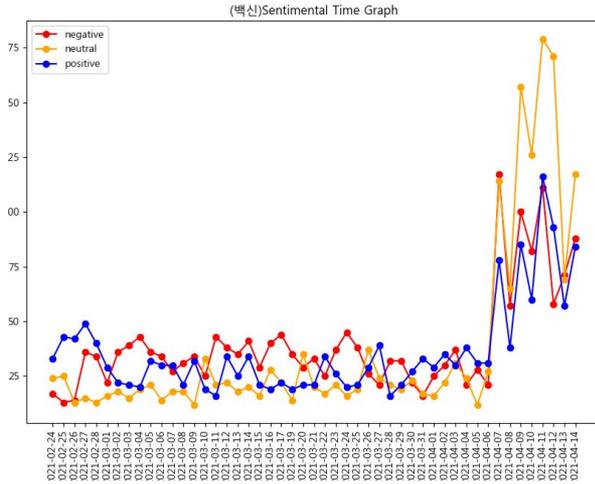
# 데이터 가공

## 키워드별 빈도수 분석

freq_AZ - Windows 메모장				freq_AZ - Windows 메모장				freq_AZ - Windows 메모장					
파일(F)	편집(E)	서식(O)	보기(V)	파일(F)	편집(E)	서식(O)	보기(V)	도움말(H)	파일(F)	편집(E)	서식(O)	보기(V)	도움말(H)
운빨:1				자다:97					백신:6467				
유추:1				입원:97					아스:4741				
번지:1				종합:97					접종:3774				
파스카:1				세계:97					맞다:2004				
소아:1				코로나바이러스:95					있다:1351				
최고경영자:1				분기:94					되다:1266				
실정:1				수출:94					화이자:1134				
상탭니:1				내일:93					코로나:1094				
주작:1				회의:93					에서:981				
후자:1				환자:92					뉴스:779				
한일:1				진자:92					이상:705				
아시:1				보고:92					없다:705				
맨뒤:1				위탁:92					이다:680				
분란:1				공장:91					효과:648				
다툼:1				사이언스:91					보다:516				
처지다:1				사실:90					생산:496				
긴밀하다:1				사망자:90					혈전:484				
래서:1				증상:89					대통령:447				
멕이기:1				일부:89					않다:446				
초도:1				이나:89					유럽:445				
버거운:1				계속:89					아니다:444				
판매승인:1				아직:87					국내:444				
기왕:1				쓰다:87					영국:423				
권영미:1				시키다:86					부터:423				
데이터베이스:1				회분:86					부작용:393				
변국:1				하나:85					한국:388				
개르:1				개발:85					국민:383				
징발:1				계약:85					출처:370				
대공황:1				중증:84					문재인:356				
소득세:1				중사:84					중단:352				
물수:1				청장:84					공급:345				
				주사기:83					하고:332				
				제외:83					까지:317				
									다음:317				
									정부:315				



# 분석 결과



시간별 부정, 중립 긍정 빈도수 변화(한글 트윗)



# 분석 결과

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm

fm.get_fontconfig_fonts()
font_location = 'C:/Windows/Fonts/malgun.ttf'
font_name = fm.FontProperties(fname=font_location).get_name()
plt.rc('font', family=font_name)

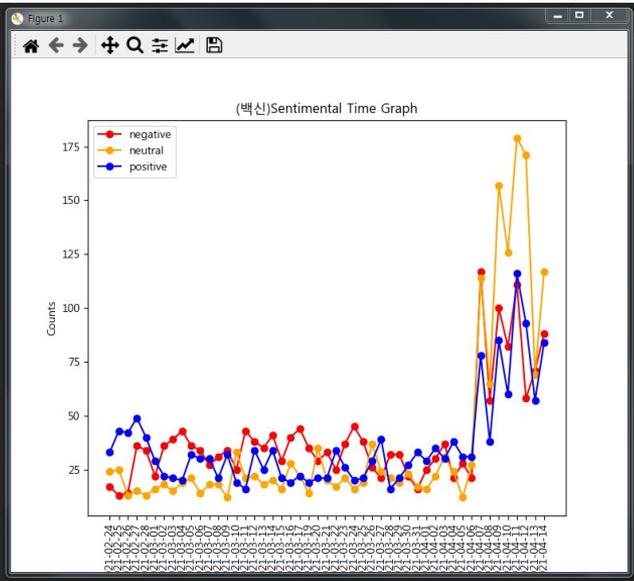
df_Vac = pd.read_excel('./data/result(백신).xlsx')
df_Vac = df_Vac.loc[:, ['date', 'Label']]

# '백신' 에 관한 timegraph
df_Vac2 = df_Vac['Label'].groupby([df_Vac['date'], df_Vac['Label']]).count()
print(type(df_Vac2))
df_Vac2 = df_Vac2.reset_index(name='count')
print(df_Vac2)

neg_df = df_Vac2.query('Label=="-1"')
days = neg_df['date']
neu_df = df_Vac2.query('Label=="0"')
pos_df = df_Vac2.query('Label=="1"')

fig = plt.figure(figsize=(10,7)) ## 캔버스 생성
fig.set_facecolor("white") ## 캔버스 색상 설정
ax = fig.add_subplot() ## 그림 해대(프레임) 생성

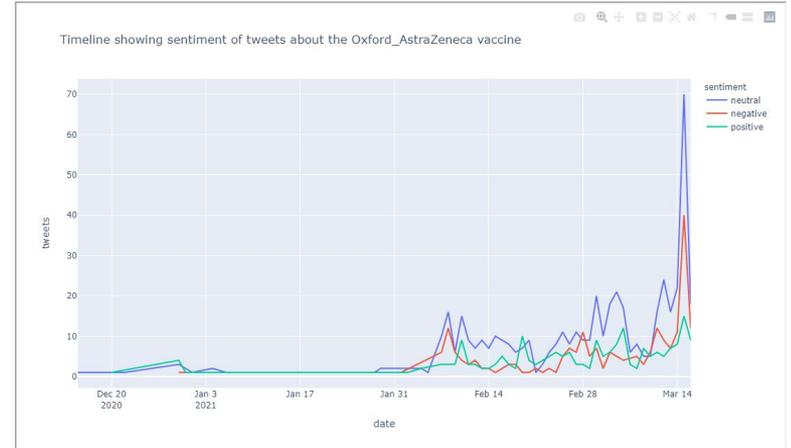
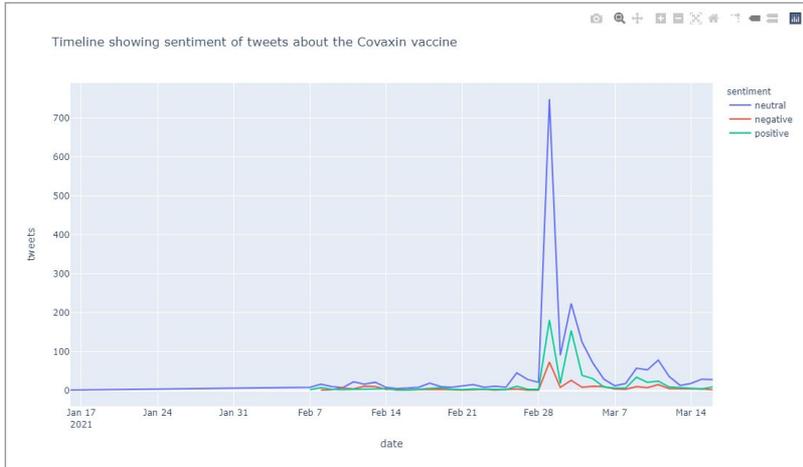
ax.plot(days, neg_df['count'], markers='o', label='negative', c='r') ## 선그래프 생성
ax.plot(days, neu_df['count'], markers='o', label='neutral', c='orange')
ax.plot(days, pos_df['count'], marker='o', label='positive', c='b')
plt.title('백신)Sentimental Time Graph')
plt.xlabel('Date')
plt.ylabel('Counts')
plt.legend()
plt.xticks(rotation=90)
plt.savefig('백신)Sentimental Time Graph')
plt.show()
```



시간별 부정, 중립 긍정 빈도수 변화(한글 트윗)



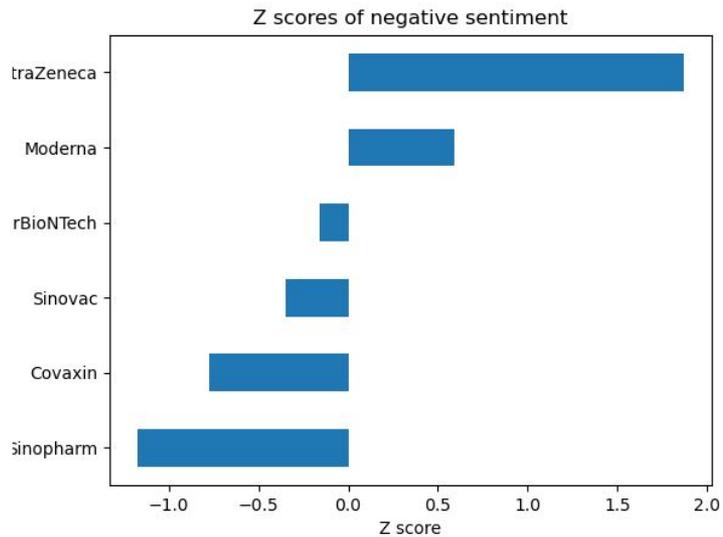
# 분석 결과



시간별 부정, 중립 긍정 빈도수 변화 (영어 트윗)



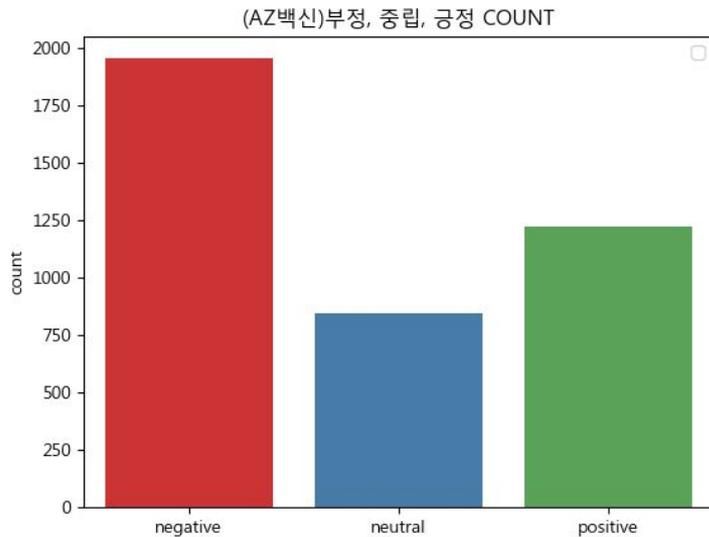
# 분석 결과



백신별 부정적인 지수 비교



# 분석 결과



부정 / 중립 / 긍정 지수비교



# 분석 결과

```
# 부정 단어
negative_data = pnn_data['Label'] == -1
negative_data = pnn_data[negative_data]
negative_data = negative_data.drop("Label", axis=1)
print(negative_data)
# 중립 단어
neutral_data = pnn_data['Label'] == 0
neutral_data = pnn_data[neutral_data]
neutral_data = neutral_data.drop("Label", axis=1)
print(neutral_data)
# 긍정 단어
positive_data = pnn_data['Label'] == 1
positive_data = pnn_data[positive_data]
positive_data = positive_data.drop("Label", axis=1)
print(positive_data)
...

#####
# 트위터 (빈도수기준) 워드클라우드
twit_coloring = np.array(Image.open('twit.png'))
if search_text == '백신':
    twit_coloring = np.array(Image.open('v_image.png'))
from wordcloud import ImageColorGenerator
image_colors = ImageColorGenerator(twit_coloring)

covid_wc = WordCloud(font_path = font_location, background_color='white',width=1000, height=500, random_state=1)

fig, ax = plt.subplots(figsize=(12,6))
plt.imshow(covid_wc, interpolation='bilinear')
plt.axis('off')
plt.title('(' + search_text + ')wordcloud')
plt.savefig('wordcloud(' + search_text + ').png')
plt.show()

...
# (은강 자체에서 데이터를 불러 올라오됨)
```



백신별 워드 클라우드



# 개선 사항

## 모델 평가(AZ)

정확도 : 0.9529120326042175

손실값 : 0.3152697682380676

## 모델 평가(PZ)

정확도 : 0.9603710174560547

손실값 : 0.2377401441335678

Ashley 애슐리는



Has 가지고 있다..  
무얼 가지고 있을까?



A great figure  
훌륭한 몸매를



영어는 결론을 먼저 내리고

나는 ...



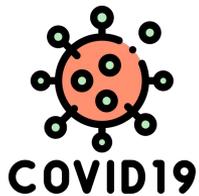
훌륭한 몸매를...



부러워한다~ ㅠ.ㅠ



우리말은 끝까지 들어보아야 안다



# FAST COVID-19 LIVE

실시간 코로나 현황을 볼 수 있는 웹을 개발해  
기존해 분석한 백신 데이터를 사이트에 접목시키기로함.



COVID-19

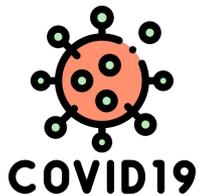
VACCIN

마지막 업데이트 : 2021-04-16 09:37:24.578

확진자	격리해제	사망자	격리중
112789	103062	1790	7937
+673	+0	+8	+1079

신규 확진자 : 해외유입 +21 / 지역발생 +652



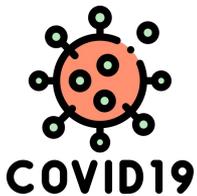


# FAST COVID-19 LIVE

총4건

<a href="#">보건의료</a> <a href="#">한국지능정보사회진흥원</a>
<b>활용신청</b> [승인] 보건복지부_코로나19 연령별·성별감염_현황
신청일 2021-04-01    만료예정일 2023-04-01
<a href="#">보건의료</a> <a href="#">한국지능정보사회진흥원</a>
<b>활용신청</b> [승인] 보건복지부_코로나19 감염_현황
신청일 2021-04-01    만료예정일 2023-04-01
<a href="#">보건의료</a> <a href="#">한국지능정보사회진흥원</a>
<b>활용신청</b> [승인] 보건복지부_코로나19 시·도발생_현황
신청일 2021-04-01    만료예정일 2023-04-01
<a href="#">보건의료</a> <a href="#">한국지능정보사회진흥원</a>
<b>활용신청</b> [승인] 보건복지부_코로나19해외발생_현황
신청일 2021-04-01    만료예정일 2023-04-01

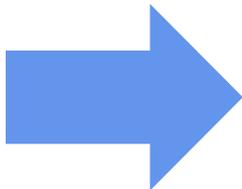
공공 데이터API를 활용해 국가별/지역별/남녀별/세대별  
감염,예방 현황을 시각화하기로함.



# FAST COVID-19 LIVE

This XML file does not appear to have any style information associated with it. The document tree is is

```
r<response>
  <header>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL SERVICE.</resultMsg>
  </header>
  <body>
    <items>
      <item>
        <createDt>2020-04-10 11:17:35.35</createDt>
        <deathCnt>0</deathCnt>
        <defCnt>352</defCnt>
        <gubun>강원</gubun>
        <gubunCh>원주</gubunCh>
        <gubunEn>Lazareto</gubunEn>
        <incDec>4</incDec>
        <isolClearCnt>3</isolClearCnt>
        <isolIngCnt>349</isolIngCnt>
        <localOccCnt>0</localOccCnt>
        <overFlowCnt>4</overFlowCnt>
        <aurRate></aurRate>
        <seq>1014</seq>
        <stdDay>2020년 04월 10일 00시</stdDay>
        <updateDt>NULL</updateDt>
      </item>
      <item>
        <createDt>2020-04-10 11:17:35.35</createDt>
        <deathCnt>0</deathCnt>
        <defCnt>12</defCnt>
        <gubun>제주</gubun>
        <gubunCh>제주</gubunCh>
        <gubunEn>Jeju</gubunEn>
        <incDec>0</incDec>
        <isolClearCnt>4</isolClearCnt>
        <isolIngCnt>8</isolIngCnt>
        <localOccCnt>0</localOccCnt>
        <overFlowCnt>0</overFlowCnt>
        <aurRate>1.79</aurRate>
        <seq>1013</seq>
        <stdDay>2020년 04월 10일 00시</stdDay>
        <updateDt>NULL</updateDt>
      </item>
      <item>
        <createDt>2020-04-10 11:17:35.35</createDt>
        <deathCnt>0</deathCnt>
        <defCnt>115</defCnt>
        <gubun>경남</gubun>
        <gubunCh>진주</gubunCh>
```



```
def get_corona_data(startCreateDt, endCreateDt):
    params = {
        'serviceKey': serviceKey_2,
        'pageNo': '1',
        'numOfRows': 10,
        'startCreateDt': startCreateDt,
        'endCreateDt': endCreateDt,
    }

    res = requests.get(url=url, params=params)
    # print(res.url)
    # print(res.text)

    # xml -> dict
    dict_data = xmltodict.parse(res.text)
    # print(dict_data)

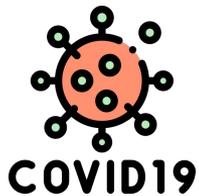
    # dict -> json
    json_data = json.dumps(dict_data)
    # print(json_data, type(json_data))

    # json -> dict
    dict_data = json.loads(json_data)
    # print(dict_data, type(dict_data))
    # pprint(dict_data['response']['body']['items']['item'])

    # total Cnt Check
    totalCount = dict_data['response']['body']['totalCount']
    if totalCount == "0":
        return False

    # 지역정보를 담은 리스트 저장
    area_data = dict_data['response']['body']['items']['item']
    area_data.reverse()
    # pprint(area_data)
    for a in area_data:
        print(a)
    return area_data
```

인증키를 받아 태그값을 추적해 원하는 데이터 추출 후  
활용하기 쉬운 형태로 가공.

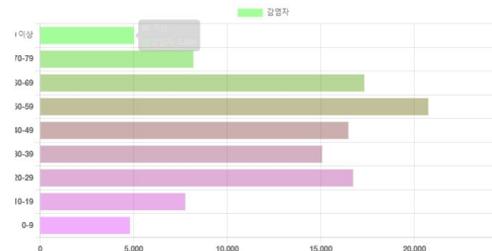


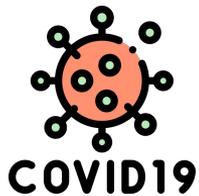
# FAST COVID-19 LIVE



## Chart.js

Chart.js를 통해 웹에서 데이터 시각화





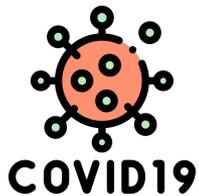
## FAST COVID-19 LIVE

This XML file does not appear to have any style information associated with it. The document tree is shown below.

---

```
▼ <response>
  ▼ <header>
    <resultCode>99</resultCode>
    <resultMsg>LIMITED NUMBER OF SERVICE REQUESTS EXCEEDS ERROR.</resultMsg>
  </header>
</response>
```

사이트의 잦은 열람 시 API 접근오류가  
걸리는 문제 발생



# FAST COVID-19 LIVE

1000

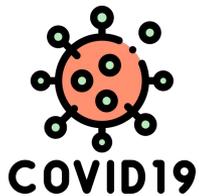
## 활용신청 상세기능정보

NO	상세기능	설명	일일 트래픽	미리보기
1	코로나19감염현황 조회 서비스	코로나19감염증으로 인한 일반 확진자, 원치자, 치료중인환자, 사망자 등에 대한 현황자료	1000	<input type="button" value="확인"/>

요청변수(Request Parameter)		모기
항목명	샘플데이터	설명
ServiceKey	-	공공데이터포털에서 받은 인증키
pageNo	1	페이지번호
numOfRows	10	한 페이지 결과 수
startCreateDt	20200310	검색할 생성일 범위의 시작
endCreateDt	20200414	검색할 생성일 범위의 종료

일일 트래픽 1000 제한이 원인.



## FAST COVID-19 LIVE

```
url = "http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19SidoInfStateJson"
url_gender = "http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19GenAgeCaseInfJson"
serviceKey_1 = '2PLDIVNdhJ3pWgmyk2qXL2LekLrwfv5r8z2tq6'
serviceKey_2 = 'h92s7QADkL1wM1LYJeYFz0x4Fmh+CV5s/VhL0j'
serviceKey_gender1= '2PLDIVNdhJ3pWgmyk2qXL2LekLrwfv5r8'
serviceKey_gender2='h92s7QADkL1wM1LYJeYFz0x4Fmh+CV5s/V'
url_vaccin = "https://nip.kdca.go.kr/ingd/cov19stats.do?list=all"
```

팀원들의 키값을 할당 받아 키오류시 다른 키를 받아 오도록  
개선.

**THANK YOU**

**THANK YOU**

**THANK YOU**

**THANK YOU**

**THANK YOU**