



نادي  
Naqi

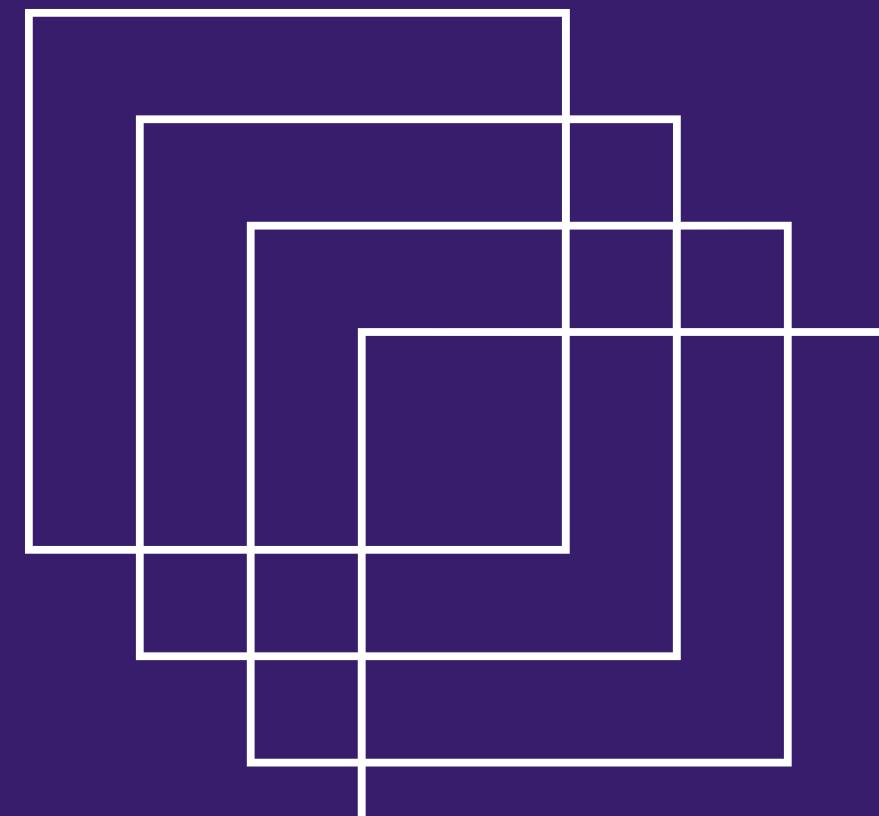


T5 -CAPSTONE PROJECT 2023

SARAH ALDAKHIL - ASEEL BIN TALHA - RAZAN ALHASSAN - RAGHAD ALADAWI

# CONTENT

- 01** WHAT IS NAQI ?
- 02** STATISTICS
- 03** METHODOLOGY
- 04** NAQI'S STORY
- 05** DEPLOY
- 06** FUTURE WORK
- 07** CONCLUSION





01

# What is Naqi ?



الهيئة العامة  
للإعلام المرئي والمسموع  
GENERAL COMMISSION  
FOR AUDIOVISUAL MEDIA

# What is Naqi ?

Naqi leverages AI to assist the Audiovisual Media Authority (GCAM) in streamlining social media advertising content monitoring, effectively combating the spread of misleading and suspicious advertisements.



02

# STATISTICS

# STATISTICS

Paid Advertising

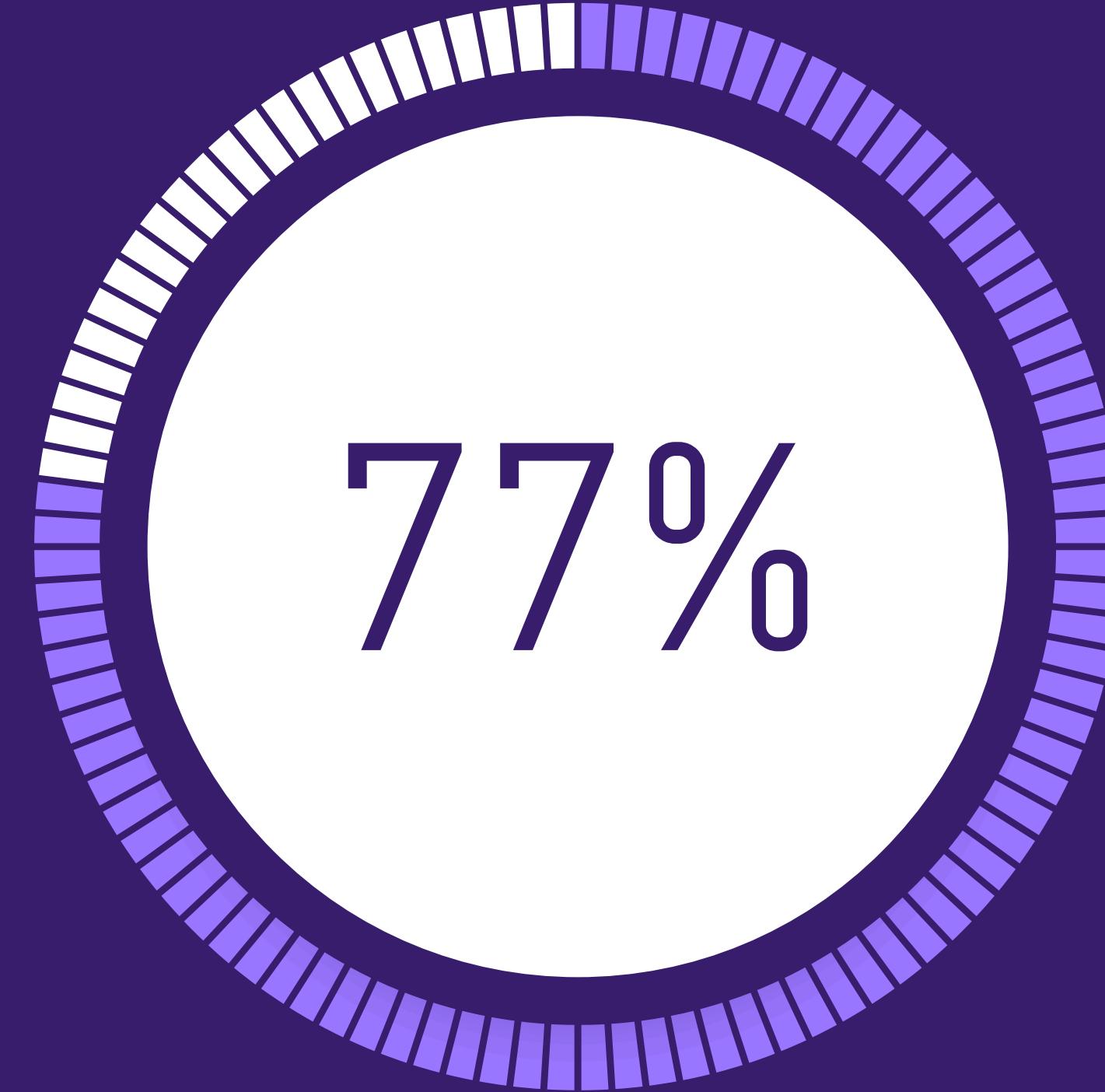
Total Ad spend

11 SR  
Billion

# STATISTICS

Paid Advertising

Digital Ad spend  
as a percentage of  
Total Ad spend



77%

A circular gauge chart with a white face and a purple outer ring divided into 100 equal segments. The number '77' is displayed in the center of the chart.

# STATISTICS

Media violations  
In the first quarter  
of 2023

4046

violation



03

## NAQI'S STORY





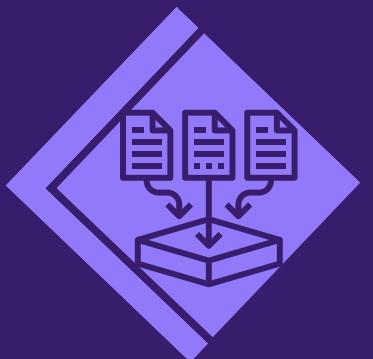
04

# METHODOLOGY

# Naqi

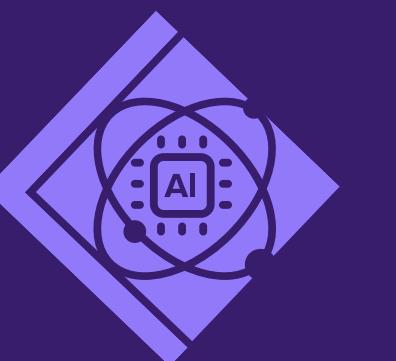
# Project Methodology

## Data collection



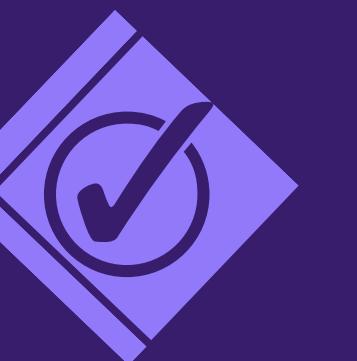
- Tweet
- Snap
- Licensed

## Model



- From Scratch
- AraBert

## Check



- Text
- Video
- Image

## NER



- Extracting company names



# 05

# DEPLOY





06

## FUTURE WORK

# FUTURE WORK

MONITORING VIOLATORS  
OF THE REMAINING TERMS



IMPLEMENTING IT FOR ALL  
SOCIAL MEDIA APPLICATIONS

EXPLORING CHILDREN  
TO PROMOTE



# 07

# CONCLUSION

# Conclusion

Naqi is a valuable initiative for the field of digital advertising, which is experiencing tremendous growth within the Kingdom of Saudi Arabia.

By working together, we can create a safe, reliable, and beneficial digital future for all Saudis.

# Thank you for listening

Any Questions ?

# Appendix

## Data Preprocessing - Stemming

```
Steem

[64] !pip install tashaphyne

Collecting tashaphyne
  Downloading Tashaphyne-0.3.6-py3-none-any.whl (251 kB)
    ━━━━━━━━━━━━━━━━━━━━ 251.5/251.5 kB 4.2 MB/s eta 0:00:00
Requirement already satisfied: pyarabic in /usr/local/lib/python3.10/dist-packages (from tashaphyne) (0.6.15)
Requirement already satisfied: six>=1.14.0 in /usr/local/lib/python3.10/dist-packages (from pyarabic->tashaphyne) (1.16.0)
Installing collected packages: tashaphyne
Successfully installed tashaphyne-0.3.6

[69] from tashaphyne.stemming import ArabicLightStemmer

stemmer = ArabicLightStemmer()

# Apply stemming to each text element in the 'text' column
FullData['text'] = FullData['text'].apply(lambda x: ' '.join([stemmer.light_stem(word) for word in x.split()]))

FullData
```

# Appendix

## Data Preprocessing - Stop Words

### Remove stopwords

```
0s  from collections import Counter

# Concatenate all text rows into a single string
all_text = ' '.join(FullData['text'])

# Tokenize the concatenated text into words
words = all_text.split()

# Count the occurrences of each word
word_counts = Counter(words)

# Get the unique words and their frequencies
unique_words = list(word_counts.keys())
word_frequencies = list(word_counts.values())

# Create a DataFrame to store the unique words and their frequencies
word_frequency_df = pd.DataFrame({'Word': unique_words, 'Frequency': word_frequencies})

# Sort the DataFrame by frequency in descending order
word_frequency_df = word_frequency_df.sort_values(by='Frequency', ascending=False)

word_frequency_df
```

```
[20] # Define the row indices of the words to be used as stopwords
stopword_indices = [4,5003,20,159,99,69,35,286,139,168,25,230,5341,5041,572,43,161]

# Get the stopwords from the 'Word' column
stopwords = list(word_frequency_df.loc[stopword_indices, 'Word'])

[21] # Function to delete words from a text
def delete_words(text, words_to_delete):
    # Split the text into words
    words = text.split()

    # Remove the words to delete
    filtered_words = [word for word in words if word not in words_to_delete]

    # Join the remaining words back into a string
    filtered_text = ' '.join(filtered_words)

    return filtered_text

# Iterate over the rows and delete words from the text column
for index, row in FullData.iterrows():
    FullData.at[index, 'text'] = delete_words(row['text'], stopwords)
```

FullData

# Appendix

## Training model - from scratch

```
# Create model3
model3 = Sequential()
model3.add(Embedding(vocab_size, embedding_size, input_length=max_length))
model3.add(SpatialDropout1D(0.2))
model3.add(Conv1D(filters=64, kernel_size=5, padding='same', activation='relu'))
model3.add(MaxPooling1D(pool_size=2))
model3.add(Bidirectional(LSTM(128, return_sequences=True, dropout=0.2, recurrent_dropout=0.2)))
model3.add(Bidirectional(LSTM(64, dropout=0.2, recurrent_dropout=0.2)))
model3.add(Reshape((-1, 128))) # Reshape the tensor to match the expected input shape for Conv
model3.add(Conv1D(filters=128, kernel_size=5, padding='same', activation='relu'))
model3.add(GlobalMaxPooling1D()) #to obtain a fixed-length representation of the sequence.
model3.add(Dense(64, activation='relu'))
model3.add(Dense(num_classes, activation='softmax'))
```

# Appendix

## Training model - from scratch

```
57s 661ms/step - loss: 0.5145 - accuracy: 0.7500 - val_loss: 0.4013 - val_accuracy: 0.7931  
57s 659ms/step - loss: 0.3186 - accuracy: 0.8574 - val_loss: 0.3114 - val_accuracy: 0.8664  
57s 660ms/step - loss: 0.1926 - accuracy: 0.9221 - val_loss: 0.2323 - val_accuracy: 0.8980  
56s 641ms/step - loss: 0.1303 - accuracy: 0.9522 - val_loss: 0.1991 - val_accuracy: 0.9339  
56s 644ms/step - loss: 0.1054 - accuracy: 0.9605 - val_loss: 0.2220 - val_accuracy: 0.9253  
57s 662ms/step - loss: 0.1070 - accuracy: 0.9601 - val_loss: 0.2837 - val_accuracy: 0.9066  
55s 630ms/step - loss: 0.0770 - accuracy: 0.9720 - val_loss: 0.2009 - val_accuracy: 0.9310
```

Test Accuracy: 0.934482753276825

# Appendix

## Data Preprocessing - Remove repeated letters + Detangling words

### Remove repeated letters

```
[ ] def remove_repeated_letters(word):
    pattern = re.compile(r'(.+)\1+')
    return pattern.sub(r'\1', word)

FullData['text'] = FullData['text'].apply(remove_repeated_letters)
FullData
```

### Detangling the word "اعلان" from the letters

```
[10] FullData['text'] = FullData['text'].str.replace(r'(\S)(اعلان)(\S)', r'\1 \2\')
```

# Appendix

## Training model - Arabert V0.2

```
Epoch 2/20
205/205 [=====] - 261s 1s/step - loss: 0.7271 - accuracy: 0.4965 - precision: 0.4715 - recall: 0.4222
Epoch 3/20
205/205 [=====] - 260s 1s/step - loss: 0.7156 - accuracy: 0.4965 - precision: 0.4702 - recall: 0.4031
Epoch 4/20
205/205 [=====] - 260s 1s/step - loss: 0.7150 - accuracy: 0.5072 - precision: 0.4824 - recall: 0.3929
Epoch 5/20
205/205 [=====] - 260s 1s/step - loss: 0.7123 - accuracy: 0.5023 - precision: 0.4758 - recall: 0.3827
Epoch 6/20
205/205 [=====] - 259s 1s/step - loss: 0.7095 - accuracy: 0.4931 - precision: 0.4659 - recall: 0.3960
Epoch 7/20
205/205 [=====] - 260s 1s/step - loss: 0.7075 - accuracy: 0.4928 - precision: 0.4660 - recall: 0.4018
Epoch 8/20
205/205 [=====] - 260s 1s/step - loss: 0.7061 - accuracy: 0.5002 - precision: 0.4750 - recall: 0.4120
Epoch 9/20
205/205 [=====] - 259s 1s/step - loss: 0.7065 - accuracy: 0.4876 - precision: 0.4575 - recall: 0.3737
Epoch 10/20
205/205 [=====] - 260s 1s/step - loss: 0.7057 - accuracy: 0.4876 - precision: 0.4586 - recall: 0.3852
Epoch 11/20
175/205 [=====>.....] - ETA: 37s - loss: 0.7012 - accuracy: 0.5000 - precision: 0.4785 - recall: 0.3873
```

# Appendix

## Training model - Few shot Multilingual

```
Epoch 1/7
Train Loss: 0.5787 | Train Accuracy: 0.6739
Validation Accuracy: 0.7543
-----
Epoch 2/7
Train Loss: 0.3184 | Train Accuracy: 0.8777
Validation Accuracy: 0.8147
-----
Epoch 3/7
Train Loss: 0.1988 | Train Accuracy: 0.9212
Validation Accuracy: 0.7112
-----
Epoch 4/7
Train Loss: 0.1683 | Train Accuracy: 0.9429
Validation Accuracy: 0.8276
-----
Epoch 5/7
Train Loss: 0.0725 | Train Accuracy: 0.9810
Validation Accuracy: 0.8017
-----
Epoch 6/7
Train Loss: 0.0628 | Train Accuracy: 0.9864
Validation Accuracy: 0.8233
-----
Epoch 7/7
Train Loss: 0.0309 | Train Accuracy: 0.9918
Validation Accuracy: 0.8147
```



Test Accuracy: 0.9207

# Appendix

## Training model - Few shot Xlm-roberta

```
Epoch 14/40
Train Loss: 0.1395 | Train Accuracy: 0.9568
Validation Accuracy: 0.8534
-----
Epoch 15/40
Train Loss: 0.1100 | Train Accuracy: 0.9600
Validation Accuracy: 0.8750
-----
Epoch 16/40
Train Loss: 0.1015 | Train Accuracy: 0.9697
Validation Accuracy: 0.8534
-----
Epoch 17/40
Train Loss: 0.0768 | Train Accuracy: 0.9784
Validation Accuracy: 0.8707
-----
Epoch 18/40
Train Loss: 0.0938 | Train Accuracy: 0.9751
Validation Accuracy: 0.8578
-----
Epoch 19/40
Train Loss: 0.0854 | Train Accuracy: 0.9795
Validation Accuracy: 0.8534
-----
Epoch 20/40
Train Loss: 0.0594 | Train Accuracy: 0.9805
Validation Accuracy: 0.8664
-----
Epoch 21/40
Train Loss: 0.0389 | Train Accuracy: 0.9924
Validation Accuracy: 0.8621
```

➡ Test Accuracy: 0.9000

# Appendix

## Data Collection - Creating Mawthooq licensed User Data set

A function that checks the User Ad license Ad (رخصة موثوق)

```
▶ import requests
import re
def check_ad_license1 (Username) :
    url = "https://twitter-api45.p.rapidapi.com/timeline.php"
    querystring = {"screenname": Username}
    headers = {
        "X-RapidAPI-Key": "ff127a487cmshd14106b9ad7f3eap12d5ffjsn9f8c75e8b3c6",
        "X-RapidAPI-Host": "twitter-api45.p.rapidapi.com"
    }

    response = requests.get(url, headers=headers, params=querystring)
    if response is None :
        return "Error"
    data = response.json()

    description = str(data['user'].get('desc', ''))  
normalized_description = description.replace("!", "").replace("?", "")
    pinned = data.get('pinned', {}).get('text', '')

    if description == 'None' and pinned == '' :
        return 'fake-account'

    if re.search(r'رخصة', normalized_description) or re.search(r'موثوق', normalized_description) or re.search(r"لإعلانات", normalized_description) or re.  
        return 'licensed'
    else:
        return 'Nonlicensed'
```

A function that get the users

```
▶ import pandas as pd
import requests

url = "https://twitter-api45.p.rapidapi.com/following.php"
querystring = {"screenname": "l_a10"}

headers = {
    "X-RapidAPI-Key": "ff127a487cmshd14106b9ad7f3eap12d5ffjsn9f8c75e8b3c6",
    "X-RapidAPI-Host": "twitter-api45.p.rapidapi.com"
}

response = requests.get(url, headers=headers, params=querystring)
data = response.json()
users = data['following']

# for user in users:
#     screen_name = user['screen_name']
#     'description' = user['description']
#     print(screen_name)

data = {'Screen Name': [], 'Description': []}

for user in users:
    data['Screen Name'].append(user['screen_name'])
    data['Description'].append(user['description'])

df = pd.DataFrame(data)
df
```

# Appendix

## Data Collection - Creating Mawthooq licensed User Data set

index	Screen Name	Description	license_status
0	iih4u	918242 لا تاخذني على محمـل الجـد - للتواصل والإعلـانات رخصـة موـثـقـة <a href="https://t.co/APE1XoeALk">https://t.co/APE1XoeALk</a>	licensed
1	_qxi7	629177 رخصـة موـثـقـة : الإعلـانات <a href="https://t.co/lxEEnrmC15">https://t.co/lxEEnrmC15</a>	licensed
2	dh_5xx	الإعلـانات <a href="https://t.co/SWSxPNqill">https://t.co/SWSxPNqill</a>	licensed
3	jussts4	133592 رخصـة موـثـقـة لـ"لا تـنـفـلـجـأـ فـقـدـ تـزـورـكـ أـفـعـالـكـ يـوـمـاـ مـاـ" الإعلـانات <a href="https://t.co/AY80jcFL7c">https://t.co/AY80jcFL7c</a>	licensed
4	20B_	"لا أـشـبـهـ وـلـأـشـبـهـ ، رخصـة موـثـقـة   الإعلـانات <a href="https://t.co/vdEcu32fNW">https://t.co/vdEcu32fNW</a>	licensed
5	Taefqlp	681915 رخصـة موـثـقـة   الإعلـانـات <a href="https://t.co/LI5K3ijxuM">https://t.co/LI5K3ijxuM</a>	licensed
6	puierq	https://t.co/U8ZKZaatEG مـسـمـوـحـ مـخـالـفـةـ رـأـيـ عـلـىـ صـفـحـتـيـ الشـخـصـيـةـ لـالـإـعـلـانـات	licensed
7	1gltx	0502260405 snapchat:mrz_4l لـالـإـعـلـانـاتـ اـسـابـ اـسـابـ <a href="https://t.co/0502260405">https://t.co/0502260405</a>	licensed
8	mrz_4l	958950 رخصـة موـثـقـة لـ"إـذـ مـرـرـتـ مـنـ هـنـاـ اـدـعـوـ لـأـيـ وـلـأـخـيـ بـالـرـحـمـةـ وـالـغـفـرـةـ" الإعلـانـاتـ وـاـسـابـ <a href="https://t.co/hYBnn8XCSf">https://t.co/hYBnn8XCSf</a>	licensed
9	true8_	شـوـشـوـ سـخـصـيـهـ مـعـرـوفـ لـالـإـعـلـانـاتـ <a href="https://t.co/0536572564">https://t.co/0536572564</a>	licensed
10	Ishyl_	362098 رخصـة موـثـقـة لـ"أـمـ زـكـيـ وـأـخـيـ الصـبـيـعـهـ هـونـ" الإعلـانـاتـ عـلـىـ <a href="https://t.co/8gNDdtaE1U">https://t.co/8gNDdtaE1U</a>	licensed
11	Arwa_0E	275155 رخصـة موـثـقـة: 275155 للـإـعـلـانـاتـ وـالـتـوـاصـلـ <a href="https://t.co/ulZcHmQFsU">https://t.co/ulZcHmQFsU</a>	licensed
12	Bashayer_w92	0564640885 أمـ زـكـيـ وـأـخـيـ الصـبـيـعـهـ هـونـ   لـالـإـعـلـانـاتـ <a href="https://t.co/Zl6HsXCwA5">https://t.co/Zl6HsXCwA5</a>	licensed
13	Nkt_767	.. إـذـ مـرـرـتـ مـنـ هـنـاـ اـدـعـوـ لـأـيـ وـلـأـخـيـ بـالـرـحـمـةـ وـالـغـفـرـةـ" <a href="https://t.co/HZCw1SEHsf">https://t.co/HZCw1SEHsf</a>	licensed
14	I_a10	44 مـلـحـةـ فـحـصـةـ نـظـمـ بـأـنـ تـصـبـحـ حـقـلـاـ <a href="https://t.co/ZS2UhybfVt">https://t.co/ZS2UhybfVt</a> .. "مـرـخصـةـ إـعلامـيـاـ" لـالـإـعـلـانـاتـ وـاـسـابـ <a href="https://t.co/0564640885">https://t.co/0564640885</a>	licensed
15	ghadii_00	50 مـصـوـرـ وـصـائـعـ مـحـتـوىـ   <a href="https://t.co/bhqY4KG4hV">https://t.co/bhqY4KG4hV</a> لـالـإـعـلـانـاتـ وـتـصـوـيرـ الـمـنـتجـاتـ <a href="https://t.co/601215">https://t.co/601215</a>	licensed
16	48Nasreen	434340 رخصـة موـثـقـة: 434340 لـالـإـعـلـانـاتـ وـالـتـعـطـيلـاتـ <a href="https://t.co/sL5MGMVuH5">https://t.co/sL5MGMVuH5</a>	licensed
17	Eja52_	805406 رخصـة موـثـقـة: 805406 لـالـإـعـلـانـاتـ <a href="https://t.co/FXQUMecVFC">https://t.co/FXQUMecVFC</a>	licensed
18	xSRx1	458249 رخصـة موـثـقـة: 458249 لـالـإـعـلـانـاتـ <a href="https://t.co/8yaTFraZFO">https://t.co/8yaTFraZFO</a>	licensed
19	Siiros2	info@hanthl.pro: مـصـصـ وـتـغـطـيـاتـ بـعـاـيـةـ فـائـقـةـ    جـمـيعـ تـغـطـيـاتـ وـالـمـكـتـوبـةـ فـيـ الإـعـجابـاتـ    لـالـإـعـلـانـاتـ أوـ الإـقـرـاحـاتـ تـوـاـصـلـواـ عـلـىـ الـخـاصـ - #مـوـثـقـةـ لـالـأـعـمـالـ	licensed
20	x2_900	267586 منـصـةـ تـرـفيـهـيـةـ - الـحـسـابـ يـمـلـكـ رـخصـةـ #مـوـثـقـةـ <a href="https://t.co/656951">https://t.co/656951</a>	licensed
21	HanthlPro	https://t.co/vUI0tkjZIR تـلـفـزـ (SK17) اـبـهـرـ (JNP607) كـوـدـ نـونـ <a href="https://t.co/vUI0tkjZIR">https://t.co/vUI0tkjZIR</a>	licensed
22	Celebrty_0	619101 لمـ يـكـنـ فـيـ هـذـاـ الـعـالـمـ شـئـ أـصـعـبـ مـنـ الصـدـقـ ... / رـخصـةـ الـإـعـلـانـاتـ : 619101	licensed
23	_Talal1		
24	salmanih0		

Show 25 per page

1 2