In this notebook I am going to download all dank images and approximately same number of non-dank images, because our original dataset is highly imbalanced thats why here I am taking approximately same number of dand and non-dank image to balance the data.

```
#importing libraries
import pandas as pd
import numpy as np
import os
import urllib.request
from tqdm import tqdm
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
#reading the dataset which contains both dank_or_not and url columns
df = pd.read_csv('/content/drive/MyDrive/Applied_ai/df_dankornot.csv')
```

```
df['dank_or_not'].value_counts()
```

```
0    81474
1     2434
Name: dank_or_not, dtype: int64
```

There are 2434 dank data.

```
#getting only dank data
df_dank = df[df['dank_or_not']==1][['url','dank_or_not']]
```

```
#selecting 2570 data from non-dank data randomly
df_non_dank = df[df['dank_or_not']==0][['url','dank_or_not']].sample(n = 2570)
```

```
#concatenating selected dank and non-dank data
df_img = pd.concat([df_dank, df_non_dank], ignore_index=True)
```

```
#making a directory with name meme_images to save the images, if the directory is
!rm -rf '/content/drive/MyDrive/Applied_ai/meme_images'
os.mkdir('/content/drive/MyDrive/Applied_ai/meme_images')
```

```
#downloading and saving images
for i in tqdm(df_img['url']):
  try:
    urllib.request.urlretrieve(i, '/content/drive/MyDrive/Applied_ai/meme_images/'
  except:
    df_img.drop(df_img.index[df_img['url']==i][0], inplace=True)
```

```
100%|██████████| 5004/5004 [11:55<00:00,  7.00it/s]
```

```
#saving the final image dataset
df_img.reset_index(drop=True).to_csv('df_img.csv', index=False)
```