In [42]:
```
#KINTALI SAHIL - CS20B1114
#PISARA THARAK - CS20B1111
```

In [8]:
```
from InstagramAPI import InstagramAPI, os
import pandas as pd
from tqdm import tqdm
import time
import numpy as np
import datetime
import networkx
import re
import matplotlib.pyplot as plt
```

In [39]:
```
def plot (xaxis, yaxis, xlabel, ylabel, title, filename):
    x = np.arange(len(xaxis))
    plt.bar(x, yaxis, color=['crimson', 'cadetblue', 'cyan', 'magenta', 'coral',
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.xticks(x, xaxis, fontsize=5, rotation=20)
    plt.title(title)
    plt.show()
    plt.savefig(filename, dpi=500)
    plt.clf()
    plt.cla()
    plt.close()

def hashtags(str):
    return re.findall(r'#(\w+)', str)
```

In [10]:
```
api = InstagramAPI("aiproject2023", "Cs20b1114") # change login and password to
time.sleep(2)
api.login()
```

```
Request return 405 error!
{'message': '', 'status': 'fail'}
Login success!
```

Out[10]:
```
True
```

In [11]:
```
# Get username info for your account
api.getSelfUsernameInfo()
user_info = api.LastJson
uid = user_info['user']['pk']
myname = user_info['user']['full_name']
```

In [12]:
```
# Grab the most recent images from everyone (friends of friends) and rate them b
# Grab the number of likes, comments and time the photo was taken
# Get the photos that you've liked to be entered in dataset used for recommendat
api.getLikedMedia()
liked_photos = api.LastJson

users_liked = [item['user'] for item in liked_photos['items']]
```

In [13]:
```
# Get the list of people following
api.getSelfUsersFollowing()
following_info = api.LastJson
```

```
follow_list=[]
for user in tqdm(following_info['users']):
    uid_followed = user['pk']
    fullname_followed = user['full_name']
    follow_list.append((uid, uid_followed, myname, fullname_followed))

    api.getUserFollowings(uid_followed)
    relationship = api.LastJson
    if relationship.get('users') is not None:
        for User in relationship['users']:
            follow_list.append((uid_followed, User['pk'], fullname_followed, Use

    time.sleep(0.5)
# Create a dataframe of follow_list
follow_list_frame = pd.DataFrame(follow_list, columns=['Source ID', 'Destination
```

```
100%|████████████████████████████████████████
████| 16/16 [00:19<00:00,  1.25s/it]
```

In [16]:

```
# Build up a network from these liked photos
user_liked_relationship = []
for user in tqdm(users_liked):
    uid_followed = user['pk']
    fullname_followed = user['full_name']
    user_liked_relationship.append((uid, uid_followed, myname, fullname_followed

    # Get the followings list for each followed user and append them
    api.getUserFollowings(uid_followed)
    user_secondary = api.LastJson

    for User in user_secondary['users']:
        user_liked_relationship.append((uid_followed, User['pk'], fullname_follo
    time.sleep(1)
```

```
100%|████████████████████████████████████████
████| 21/21 [00:33<00:00,  1.60s/it]
```

In [17]:

```
# Create a dataframe from the liked photos list
liked_list_frame = pd.DataFrame(user_liked_relationship, columns=['Source ID', '

#Change end index for a larger dataframe keeping in mind what's the maximum
liked_list_frame = liked_list_frame.iloc[0:500]
unique_user_id = np.unique(liked_list_frame[['Source ID', 'Destination ID']].val
```

In [18]:

```
# Create a graph to calculate relative score of pages
graph = networkx.from_pandas_edgelist(liked_list_frame, 'Source ID', 'Destinatic
personalised_rank = dict(zip(graph.nodes(), [0] * len(graph.nodes())))
personalised_rank[uid] = 1
pg_rank = networkx.pagerank(graph, personalization=personalised_rank)
```

In [19]:

```
# Creating dataset for providing recommendations
total_urls = []
timetaken = []
number_of_likes = []
number_of_comments = []
pagerank = []
users = []
```

In [20]:

```
for id in tqdm(unique_user_id):
    api.getUserFeed(id)
    feed = api.LastJson

    if 'items' in feed.keys():
        for item in feed['items']:

            #Capture only images for this project
            if 'image_versions2' in item.keys():
                url = item['image_versions2']['candidates'][0]['url']
                taken_at = item['taken_at']
                # Check how many likes are provided with except block if 0 likes
                try:
                    like = item['like_count']
                except KeyError:
                    like = 0

                # Check the number of comments and provide an except block if ze
                try:
                    comment = item['comment_count']
                except KeyError:
                    comment = 0

                #rank = pg_rank[item['user']['pk']]
                if item['user']['pk'] in pg_rank:
                    rank = pg_rank[item['user']['pk']]
                else:
                    rank = 0 # or some other default value

                user = item['user']['full_name']

                # dont count oneself when appending
                if user != myname:
                    total_urls.append(url)
                    timetaken.append(taken_at)
                    number_of_likes.append(like)
                    number_of_comments.append(comment)
                    pagerank.append(rank)
                    users.append(user)

      #time.sleep(1)
```

```
 41%|████████████████▊
| 32/79 [00:39<00:44,  1.07it/s]
Request return 400 error!
{'message': 'Not authorized to view user', 'status': 'fail'}
 44%|██████████████████▏
| 35/79 [00:42<00:37,  1.17it/s]
Request return 400 error!
{'message': 'Not authorized to view user', 'status': 'fail'}
 75%|██████████████████████████████▌
| 59/79 [01:09<00:18,  1.05it/s]
Request return 400 error!
{'message': 'Not authorized to view user', 'status': 'fail'}
 84%|██████████████████████████████████
| 66/79 [01:17<00:12,  1.03it/s]
Request return 400 error!
{'message': 'Not authorized to view user', 'status': 'fail'}
```

```
100%|████████████████████████████████████████████████████████████████████████████████
████████| 79/79 [01:29<00:00,  1.13s/it]
```

In [21]:
```
# With the dataset availbale, create a dataframe from it
relative_score = pd.DataFrame(
   {
      'URLS': total_urls,
      'Time taken': timetaken,
      'Number of likes': number_of_likes,
      'Number of comments': number_of_comments,
      'Rank': pagerank,
      'Users': users,
   }
)
```

In [22]:
```
# Find out the score
# A simple way to do it is just multiply to multiply parameters # of comments, #
# This simple score is very big and therefore we can standardize it by taking th
relative_score['Total Score'] = np.log(relative_score['Number of likes']) * np.l
```

```
C:\Users\hp\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\co
re\arraylike.py:397: RuntimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

In [23]:
```
# Get the top number of highest rated posts. I user 10 of these posts
relative_score = relative_score.sort_values(by='Total Score', ascending=False)
recommended_url = relative_score['URLS'].tolist()
recommended_post = relative_score['Total Score'].tolist()

recommended_url = recommended_url[0:10]
recommended_post = recommended_post[0:10]
```

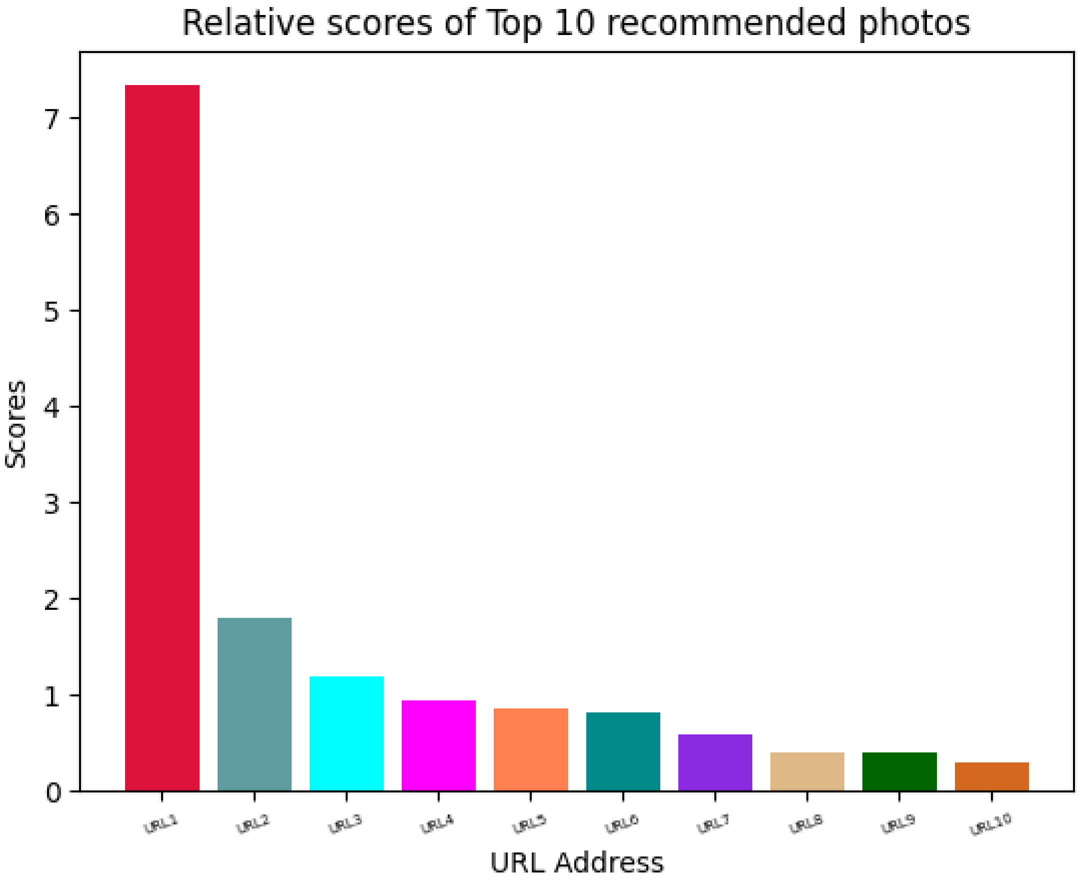
In [24]:
```
# providing urls keys so that they can be easily shown on the bar chart
indexes = ['URL1', 'URL2', 'URL3', 'URL4', 'URL5', 'URL6', 'URL7', 'URL8', 'URL9
url_dict = {}

for i in range(10):
    url_dict[indexes[i]] = recommended_url[i]

#prinitng dictionary
print("\n\nURLS of Top 10 recommended pictures")
print('--------------------------------------')
for key, values in url_dict.items():
    print(key + '-->' + values + "\n")
print("\n\n")
```

```
URLS of Top 10 recommended pictures
------------------------------------
URL1-->https://instagram.fmaa2-1.fna.fbcdn.net/v/t51.2885-15/341000246_14156623
8854504_3528172723972681219_n.webp?stp=dst-jpg_e35_p640x640_sh0.08&_nc_ht=insta
gram.fmaa2-1.fna.fbcdn.net&_nc_cat=1&_nc_ohc=xjFSi8FTMdUAX9GfZLp&edm=ABmJApABAA
AA&ccb=7-5&ig_cache_key=MzA4MTQwMTExMzAzOTkwMzgzMA%3D%3D.2-ccb7-5&oh=00_AfBcMby
rTDqw4gAvdwQK4x-9RcnvHZc2fJVM5om_mGUrJw&oe=643FEDF6&_nc_sid=6136e7

URL2-->https://instagram.fmaa2-1.fna.fbcdn.net/v/t51.2885-15/341162877_76521835
4991558_6649808307619622976_n.jpg?stp=dst-jpg_e35_p640x640_sh0.08&_nc_ht=instag
ram.fmaa2-1.fna.fbcdn.net&_nc_cat=1&_nc_ohc=qMsP7BvGyzkAX9TDUCy&edm=ABmJApABAAA
A&ccb=7-5&ig_cache_key=MzA4MTYyMTU5MzQ3MzYzNDI0Mg%3D%3D.2-ccb7-5&oh=00_AfDqgr38
SkLGZqrfGdIefXwU1Mwe1Dt5UoksxdIjvM5n-Q&oe=6440C1C3&_nc_sid=6136e7

URL3-->https://instagram.fmaa2-1.fna.fbcdn.net/v/t51.2885-15/341164930_59008844
6422311_8637459835369615546_n.jpg?stp=dst-jpg_e35&_nc_ht=instagram.fmaa2-1.fna.
fbcdn.net&_nc_cat=1&_nc_ohc=OeSn-_c-LcYAX8Zsh_m&edm=ABmJApABAAAA&ccb=7-5&ig_cac
he_key=MzA4MTM3MTI4MzUzMjEwMTEzOA%3D%3D.2-ccb7-5&oh=00_AfDGfAGmBskBcnZZOz9SANTh
xQdyknoR-DOBJdEFoS_QxA&oe=6441370D&_nc_sid=6136e7

URL4-->https://instagram.fmaa2-1.fna.fbcdn.net/v/t51.2885-15/340806940_17111875
5836767_4509870944194903055_n.jpg?stp=dst-jpg_e35_p640x640_sh0.08&_nc_ht=instag
ram.fmaa2-1.fna.fbcdn.net&_nc_cat=1&_nc_ohc=wgrjF0WE_w8AX-WQyjZ&edm=ABmJApABAAA
A&ccb=7-5&ig_cache_key=MzA3OTUzNjQ1Mzg1OTk2NDIwOA%3D%3D.2-ccb7-5&oh=00_AfCR2MqC
dxpJG1sicdkaT2skki0-UFnBFBNN49tOkMy1IA&oe=6440AD1C&_nc_sid=6136e7

URL5-->https://instagram.fmaa2-1.fna.fbcdn.net/v/t51.2885-15/341335040_35097358
72680112_9198290424622623454_n.jpg?stp=dst-jpg_e15_s240x240&_nc_ht=instagram.fm
aa2-1.fna.fbcdn.net&_nc_cat=1&_nc_ohc=nsorLhwXZtQAX-6l7Iq&edm=ABmJApABAAAA&ccb=
7-5&ig_cache_key=MzA4MTM5MzQ1MDYwNDkzNDU1MA%3D%3D.2-ccb7-5&oh=00_AfDxF8SOVhUvIt
flldasYpHHmZHaWkqsyYkygdP_dX6kwQ&oe=64408100&_nc_sid=6136e7

URL6-->https://instagram.fmaa2-1.fna.fbcdn.net/v/t51.2885-15/340983659_12355326
63733812_2207470147040379075_n.jpg?stp=dst-jpg_e35_s640x640_sh0.08&_nc_ht=insta
gram.fmaa2-1.fna.fbcdn.net&_nc_cat=1&_nc_ohc=HKGdg6SO8d8AX-MYhM3&edm=ABmJApABAA
AA&ccb=7-5&ig_cache_key=MzA4MTM3NTQ1MjIzNzI3MTc0Ng%3D%3D.2-ccb7-5&oh=00_AfDOXN7
f-_kaNZJ7stJtjKIsbPvxYheCog3XBtLI64feGw&oe=643FD126&_nc_sid=6136e7

URL7-->https://instagram.fmaa2-2.fna.fbcdn.net/v/t51.2885-15/330126262_26581603
5780671_2578233304059746279_n.jpg?stp=dst-jpg_e35_p640x640_sh0.08&_nc_ht=instag
ram.fmaa2-2.fna.fbcdn.net&_nc_cat=111&_nc_ohc=E1lY28Hl-V8AX-FJM0p&edm=ABmJApABA
AAA&ccb=7-5&ig_cache_key=MzA4MTk5MTAyMTY3MDkyODAxOA%3D%3D.2-ccb7-5&oh=00_AfCKKU
Um1NGuwZhmQ4nIv-v1vG8JxxNA3kUwPntzpH6NTg&oe=6441153C&_nc_sid=6136e7

URL8-->https://instagram.fmaa2-1.fna.fbcdn.net/v/t51.2885-15/341148143_13760003
66275537_6779666883266855475_n.jpg?stp=dst-jpg_e15_p240x240&_nc_ht=instagram.fm
aa2-1.fna.fbcdn.net&_nc_cat=1&_nc_ohc=cMF-q5jJTLcAX9S0CWz&edm=ABmJApABAAAA&ccb=
7-5&ig_cache_key=MzA4MDQ5NDAwOTY2ODQwMjczMg%3D%3D.2-ccb7-5&oh=00_AfBjLDmpDr_1Iu
FuIL1xHOpBMf4HsKN2npD-kavRpsWfMg&oe=64415543&_nc_sid=6136e7

URL9-->https://instagram.fmaa2-1.fna.fbcdn.net/v/t51.2885-15/341560713_47450279
78955313_5511028488446427486_n.jpg?stp=dst-jpg_e35_p640x640_sh0.08&_nc_ht=insta
gram.fmaa2-1.fna.fbcdn.net&_nc_cat=1&_nc_ohc=Cp6gKC4xNVsAX9dNINL&edm=ABmJApABAA
AA&ccb=7-5&ig_cache_key=MzA4MDc1MTU2MTE2MTEyMDAwMA%3D%3D.2-ccb7-5&oh=00_AfAReXI
fWGDFtgPHJM6YiQXcBvV69fV_mDVDLOWV_rStkw&oe=64405BCB&_nc_sid=6136e7

URL10-->https://instagram.fmaa2-1.fna.fbcdn.net/v/t51.2885-15/340999414_9769070
50350396_1351123365697880172_n.jpg?stp=dst-jpg_e35_p640x640_sh0.08&_nc_ht=insta
gram.fmaa2-1.fna.fbcdn.net&_nc_cat=1&_nc_ohc=fEhumFcY4g8AX-hXhz0&edm=ABmJApABAA
```

```
AA&ccb=7-5&ig_cache_key=MzA4MDY1NDMwNzYzMTg3MjI5Mg%3D%3D.2-ccb7-5&oh=00_AfDw6nH
xTIxp1AkWJtBbwLcfHxSJm87tGJRIXUJK_udpxA&oe=643FE0D4&_nc_sid=6136e7
```

In [40]:

```python
#Visualization
plot(indexes, recommended_post, "URL Address", "Scores", "Relative scores of Top
```



In [26]:

```python
# Find the images according to the hashtags a user mostly likes
# Very important when considering the tastes of travel enthusiasts
total_hashtags = []

# Get recently liked photos
api.getLikedMedia()
liked_photos = api.LastJson
for item in tqdm(liked_photos['items']):
    if item['caption'] is not None:
        hashtag = hashtags(item['caption']['text'])
        [total_hashtags.append(i.lower()) for i in hashtag]

   #time.sleep(1)
```

```
100%|█████████████████████████████████████████████████████
████████████| 21/21 [00:00<?, ?it/s]
```

In [27]:

```python
recommended_hashtags = pd.Series(total_hashtags).value_counts()[:10]
top_hashtag_images = {}

for hashtag in recommended_hashtags.index:
    api.getHashtagFeed(hashtag)
    feed = api.LastJson
    top_hashtag_images[hashtag] = feed
    #time.sleep(1)
```

In [28]:

```
total_urls = []
number_of_likes = []
tags = []
number_of_comments = []
```

In [30]:

```
for hashtag in top_hashtag_images.keys():
    images = top_hashtag_images[hashtag]['items']
    for item in images:
        if 'image_versions2' in item:
            total_urls.append(item['image_versions2']['candidates'][0]['url'])
            # Check how many likes are provided with except block if 0 likes
            try:
                number_of_likes.append(item['like_count'])
            except KeyError:
                number_of_likes.append(0)

            # Check the number of comments and provide an except block if zero c
            try:
                number_of_comments.append(item['comment_count'])
            except KeyError:
                number_of_comments.append(0)

            tags.append(hashtag)

   #time.sleep(1)
```

In [31]:

```
tag_frame = pd.DataFrame(
   {
      'Tags': tags,
      'Total urls': total_urls,
      'Number of likes': number_of_likes,
      'Number of comments': number_of_comments
   }
)

tag_frame['Score'] = np.log(tag_frame['Number of likes'] * tag_frame['Number of
popular_tags = tag_frame.groupby('Tags').max()
popular_tags = popular_tags.sort_values('Score', ascending=False)
```

```
C:\Users\hp\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\co
re\arraylike.py:397: RuntimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

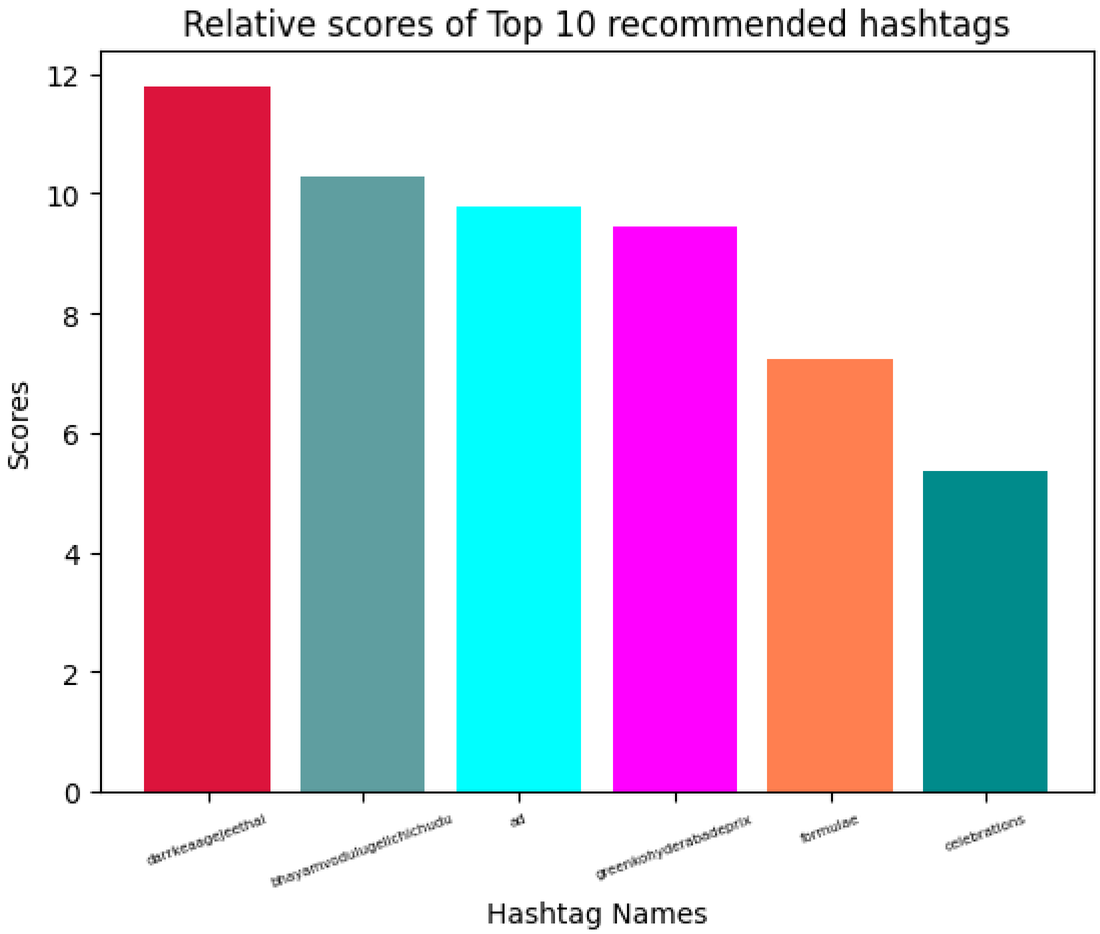
In [32]:

```
count = 0
list_name = []
for a, b in popular_tags.iterrows():
    count = count + 1
    list_url = b.tolist()
    list_name.append(a)
    list_url = list_url[:len(list_url) - 3]
    print(f"Recommendation number {count} is: " + a)
    print(list_url)
    print("\n\n")
```

```
Recommendation number 1 is: darrkeaagejeethai
['https://instagram.fmaa2-3.fna.fbcdn.net/v/t51.2885-15/341018246_1268875020371
212_4157835276529771449_n.jpg?stp=dst-jpg_e15_p240x240&_nc_ht=instagram.fmaa2-
3.fna.fbcdn.net&_nc_cat=104&_nc_ohc=TLBr1OwOXzgAX-Ctg3B&edm=AMT5gP0BAAAA&ccb=7-
5&ig_cache_key=MzA3OTkxNjk1MTc1NjkzMTgwMQ%3D%3D.2-ccb7-5&oh=00_AfAzi_YQpFZF8i0U
Nfs6f-nU9ZACj23-0jDEtTHbjLJKJw&oe=6440C6D7&_nc_sid=d50af0']



Recommendation number 2 is: bhayamvodulugelichichudu
['https://instagram.fmaa2-3.fna.fbcdn.net/v/t51.2885-15/334445663_1256036458679
301_4101882047439449968_n.jpg?stp=dst-jpg_e35_s640x640_sh0.08&_nc_ht=instagram.f
maa2-3.fna.fbcdn.net&_nc_cat=109&_nc_ohc=9hSB4jTmyZoAX99DlrP&edm=AMT5gP0BAAAA&c
cb=7-5&ig_cache_key=MzA1MTY4NTIxNDAwNzU5NjA4MA%3D%3D.2-ccb7-5&oh=00_AfCcC8warvZ
ztUnlVnMKF0B4kjjSGJzKROEg08fXS8s6Rw&oe=643FAB54&_nc_sid=d50af0']



Recommendation number 3 is: ad
['https://instagram.fmaa2-3.fna.fbcdn.net/v/t51.2885-15/341556810_1489677961439
037_1691282924147273165_n.jpg?stp=dst-jpg_e15_p240x240&_nc_ht=instagram.fmaa2-
3.fna.fbcdn.net&_nc_cat=109&_nc_ohc=TGdl8kyGCe8AX-yI7jx&edm=AMT5gP0BAAAA&ccb=7-
5&ig_cache_key=MzA4MjAzNjg5NDM4NTcyNjgwMw%3D%3D.2-ccb7-5&oh=00_AfD6Pw8gDXZASDnu
5huVG9eClIUwCqp6VwbhHg5FGCZssg&oe=64414E31&_nc_sid=d50af0']



Recommendation number 4 is: greenkohyderabadeprix
['https://instagram.fmaa2-3.fna.fbcdn.net/v/t51.2885-15/336003497_1196167584359
572_133910906260296812_n.webp?stp=dst-jpg_e35_s640x640_sh0.08&_nc_ht=instagram.
fmaa2-3.fna.fbcdn.net&_nc_cat=104&_nc_ohc=5y1xHt8ZJG8AX-0MojY&edm=AMT5gP0BAAAA&
ccb=7-5&ig_cache_key=MzA1OTAzNDU3MzQ4MzkwNjEzNw%3D%3D.2-ccb7-5&oh=00_AfDTMHgkKG
4iMw2IiaGyMxNOnmTP62i3iCmwU5Xm-WmtIg&oe=6441381B&_nc_sid=d50af0']



Recommendation number 5 is: formulae
['https://instagram.fmaa2-3.fna.fbcdn.net/v/t51.2885-15/341909034_7744817507811
52_5831318915239112958_n.jpg?stp=dst-jpg_e15_p240x240&_nc_ht=instagram.fmaa2-3.
fna.fbcdn.net&_nc_cat=104&_nc_ohc=hpRGoFuJ3gQAX82tMsr&edm=AMT5gP0BAAAA&ccb=7-5&
ig_cache_key=MzA4MTYzMTcyMzA0NDQxNzk5NQ%3D%3D.2-ccb7-5&oh=00_AfBgAoKkS-tuFd9G-O
wP4cbI4uDno5ZMYrwqbM8Ojc3EVA&oe=6440378A&_nc_sid=d50af0']



Recommendation number 6 is: celebrations
['https://instagram.fmaa2-3.fna.fbcdn.net/v/t51.2885-15/341579810_9418924269447
76_5514784302843729018_n.jpg?stp=dst-jpg_e15_p240x240&_nc_ht=instagram.fmaa2-3.
fna.fbcdn.net&_nc_cat=104&_nc_ohc=RwJaWEbDGxAAX8-3GZ3&edm=AMT5gP0BAAAA&ccb=7-5&
ig_cache_key=MzA4MjAyMTAyNjM4ODAwMjQ2NQ%3D%3D.2-ccb7-5&oh=00_AfBGXyochwspdFKR-U
ozCzYuDGCoILzNEk7z4LrTUKC1OA&oe=6440623D&_nc_sid=d50af0']
```

In [41]:

```
# Visualization
list_score = popular_tags['Score'].tolist()
plot(list_name, list_score, "Hashtag Names", "Scores", "Relative scores of Top 1
```


Relative scores of Top 10 recommended hashtags
Scores
12
10
8
6
4
2
0
darrkeaagejeethai
bhayamvodulugelichichudu
ad
greenkohyderabadeprix
formulae
celebrations
Hashtag Names

```
In [ ]:
```