```
#!/usr/bin/env python
# coding: utf-8

# In[2]:


from bs4 import BeautifulSoup
import requests
import numpy as np
import pandas as pd
import warnings

warnings.filterwarnings("ignore")


# In[3]:


def get_listings(api_key, listing_url):
    url = "https://app.scrapeak.com/v1/scrapers/zillow/listing"

    querystring = {
        "api_key": api_key,
        "url":listing_url
    }

    return requests.request("GET", url, params=querystring)

def get_property_detail(api_key, zpid):
    url = "https://app.scrapeak.com/v1/scrapers/zillow/property"

    querystring = {
        "api_key": api_key,
        "zpid":zpid
    }

    return requests.request("GET", url, params=querystring)

def get_zpid(api_key, street, city, state, zip_code=None):
    url = "https://app.scrapeak.com/v1/scrapers/zillow/zpidByAddress"

    querystring = {
        "api_key": api_key,
        "street": street,
        "city": city,
        "state": state,
        "zip_code":zip_code
    }

    return requests.request("GET", url, params=querystring)


# In[4]:


api_key = "59e77573-2bc8-482b-96ea-90d******"


# In[5]:


listing_url = "https://www.zillow.com/homes/for_sale/?searchQueryState=%7B%22usersSearchTerm%22%3A%22Tampa%2C%20FL%22%2C%22mapBounds%22%3A%7B%22north%22%3A39.0

listing_response = get_listings(api_key, listing_url)


# In[6]:


num_of_properties = listing_response.json()["data"]["categoryTotals"]["cat1"]["totalResultCount"]
print("Count of properties:", num_of_properties)


# In[7]:


df_listings = pd.json_normalize(listing_response.json()["data"]["cat1"]["searchResults"]["mapResults"])
print("Number of rows:", len(df_listings))
print("Number of columns:", len(df_listings.columns))
df_listings.iloc[:,90:]


# In[15]:


df_listings.columns


# In[8]:


df_listings.head()


# In[16]:


df = df_listings.iloc[:,36:49]


# In[17]:


df.head()
```

```
# In[10]:


df.drop(df.columns[:2],axis=1,inplace=True)


# In[11]:


df.head()


# In[36]:


df2 = df_listings.iloc[:,50:68]


# In[37]:


df2.head()


# In[38]:


df_final = pd.concat([df,df2],axis=1)


# In[39]:


df_final.shape


# In[40]:


df_final.head()


# In[44]:


df_final.columns


# In[45]:


new_names = {
    'hdpData.homeInfo.zpid':'zpid',
    'hdpData.homeInfo.streetAddress':'Address',
    'hdpData.homeInfo.zipcode':'zipcode',
    'hdpData.homeInfo.city':'city',
    'hdpData.homeInfo.state':'state',
    'hdpData.homeInfo.latitude':'latitude',
    'hdpData.homeInfo.longitude':'longitude',
    'hdpData.homeInfo.price':'price',
    'hdpData.homeInfo.bathrooms':'bathrooms',
    'hdpData.homeInfo.bedrooms':'bedrooms',
    'hdpData.homeInfo.currency':'currency',
    'hdpData.homeInfo.country':'country',
    'hdpData.homeInfo.taxAssessedValue':'taxAssessedvalue',
    'hdpData.homeInfo.lotAreaValue':'lotAreaValue',
    'hdpData.homeInfo.lotAreaUnit':'lotAreaUnit',
    'hdpData.homeInfo.livingArea':'livingArea',
    'hdpData.homeInfo.homeType':'homeType',
    'hdpData.homeInfo.homeStatus':'homeStatus',
    'hdpData.homeInfo.listing_sub_type.is_bankOwned':'is_bankOwned',
    'hdpData.homeInfo.isUnmappable':'isUnmappable',
    'hdpData.homeInfo.isPreforeclosureAuction':'isPreforeclosureAuction',
    'hdpData.homeInfo.isNonOwnerOccupied':'isNonOwnerOccupied',
    'hdpData.homeInfo.isPremierBuilder':'isPremierBuilder',
    'hdpData.homeInfo.isZillowOwned':'isZillowOwned'


}
df_final = df_final.rename(columns=new_names)


# In[42]:


df_final.drop(columns=['hdpData.homeInfo.isFeatured','hdpData.homeInfo.shouldHighlight','hdpData.homeInfo.zestimate','hdpData.homeInfo.rentZestimate','hdpData.


# In[46]:


df_final.head()


# In[47]:


df_final.dtypes


# In[48]:


df_final['zipcode'] = df_final['zipcode'].astype('int')
df_final['city'] = df_final['city'].astype('string')
```

```
df_final['state'] = df_final['state'].astype('string')
df_final['bathrooms'] = df_final['bathrooms'].astype('int')
df_final['bedrooms'] = df_final['bedrooms'].astype('int')
df_final['homeType'] = df_final['homeType'].astype('string')
df_final['currency'] = df_final['currency'].astype('string')
df_final['country'] = df_final['country'].astype('string')
df_final['lotAreaUnit'] = df_final['lotAreaUnit'].astype('string')



# In[49]:


df_final.head()


# In[50]:


df_final.to_csv('real_estate_data.csv',index=False)


# In[ ]:
```