Data Scrapping for Delhi - Paint Dealers from multiple brands from the website :

"https://www.dealerservicecenter.in/ (https://www.dealerservicecenter.in/)"

In [ ]:

```
import pandas as pd
from bs4 import BeautifulSoup
import requests
from selenium import webdriver
import time
import re
import urllib.parse
import pandas as pd
import json
import math
from geopy import geocoders
from geopy.geocoders import Nominatim
```

In [ ]:

```
options = webdriver.ChromeOptions()
driver = webdriver.Chrome(executable_path=r'<Path to your chromedriver>/chromedriver.exe')

name1=[]
address1=[]
contact1=[]
dealers1=[]

paint_dealers=['asian-paints','british-paints','dulux','kansai-nerolac','nippon-paint','jotun','jsw-paints','shalimar','mrf','indigo-paints']

for i in range(0,len(paint_dealers)):
    url = "https://www.dealerservicecenter.in/list/paint/{}/delhi/delhi".format(paint_dealers[i])

    driver.get(url)
    r = requests.get(url)
    soup = BeautifulSoup(r.content, 'lxml')
    table = soup.findAll('div',{'class':'panel panel-default testimonial'})
    print(url)


    for j in range(5,len(table)):
        try:
            name = driver.find_element_by_xpath("//*[@id='content']/div/div/div[1]/div/div/div/div[{}]/div/h3/a".format(j)).text
            address = driver.find_element_by_xpath("//*[@id='content']/div/div/div[1]/div/div/div/div[{}]/div/ul/li[1]".format(j)).text
            contact = driver.find_element_by_xpath("//*[@id='content']/div/div/div[1]/div/div/div/div[{}]/div/ul/li[3]".format(j)).text
            name1.append(name)
            address1.append(address)
            contact1.append(contact)
            dealers1.append(paint_dealers[i])
        except Exception as ex:
            name = driver.find_element_by_xpath("//*[@id='content']/div/div/div[1]/div/div/div/div[{}]/div/h3/a".format(j+1)).text
            address = driver.find_element_by_xpath("//*[@id='content']/div/div/div[1]/div/div/div/div[{}]/div/ul/li[1]".format(j+1)).text
            contact = driver.find_element_by_xpath("//*[@id='content']/div/div/div[1]/div/div/div/div[{}]/div/ul/li[3]".format(j+1)).text
            name1.append(name)
            address1.append(address)
            contact1.append(contact)
            dealers1.append(paint_dealers[i])
            print(ex)
```

In [ ]:

```
final_data=pd.DataFrame(list(zip(name1,address1,contact1,dealers1)),columns=['Name','Address','Contact','Dealer'])
len(final_data)
```

By running this script you will get the data for Delhi Paint Dealers for the brands mentioned above

In [ ]:

```
final_data.drop_duplicates(inplace=True)
final_data=pd.DataFrame(final_data)
final_data.reset_index(inplace=True)
final_data.drop(columns=['index'],axis=1,inplace=True)
final_data.head()
```

In [ ]:

```
final_data['Address'] = final_data['Address'].map(lambda x: x.lstrip('Address:\n'))
final_data['Contact'] = final_data['Contact'].map(lambda x: x.lstrip('Contact Number:\n
').rstrip(' REPORT WRONG NUMBER!'))

final_data.head()
```

In [ ]:

```
final_data['Pincode']=pd.DataFrame(final_data,columns=['Pincode'])
final_data['State']=pd.DataFrame(final_data,columns=['State'])

for i in range(0,len(final_data)):
    final_data['State']='Delhi'
final_data.head(10)
```

In [ ]:

```
a=[]
str=final_data['Address']

for i in str:

    if re.findall(r'- (\d{6})', i):
        x=re.findall(r'- (\d{6})', i)
        if x[0] in i:
            start_index = i.index(x[0])
            s= i[start_index:start_index+6]
            a.append(s)

    else :
        a.append('--')


final_data['Pincode'] = pd.DataFrame(a)
final_data
```

In [ ]:

```
data1=pd.DataFrame(final_data)

pincode=data1['Pincode']

from geopy.geocoders import Nominatim
geolocator = Nominatim(user_agent='my_appli')


for i in pincode:
    try:
        if i is not None :
            location = geolocator.geocode(i)
            #data1['lat'].append(location.latitude)

            #print(location.address)
            print((location.latitude,location.longitude))
    except:
        print('- -')
data1.insert(5, 'latitude', location.latitude, allow_duplicates=False)
data1.insert(6, 'longitude', location.longitude, allow_duplicates=False)
```

In [ ]:

```
data1.to_excel("<Path where you want to save the file>/<Filename>.xlsx",index=False)
```