Amazon web scrapper project

June 30, 2023

[1]:

```
# Import libraries
from bs4 import BeautifulSoup
import requests
import smtplib
import time
import datetime
```

[2]:

```
import requests
from bs4 import BeautifulSoup

URL = 'https://www.amazon.in/Apple-iPhone-14-128GB-Starlight/dp/B0BDK8LKPJ/
 ↪ref=sr_1_4?
 ↪crid=Y2Q6HTL0UEY3&keywords=iphone+13&qid=1687979356&sprefix=iphone+13%2Caps%2C284&sr=8-4'

# Send a GET request to the URL
response = requests.get(URL)

# Parse the HTML content
soup = BeautifulSoup(response.content, 'html.parser')

# Find the title and price elements using appropriate selectors
title_element = soup.find('span', attrs={'id': 'productTitle'})
price_element = soup.find('span', attrs={'class': 'a-price-whole'})
discount_element = soup.find('span', attrs={'class': 'a-size-large␣
 ↪a-color-price savingPriceOverride aok-align-center␣
 ↪reinventPriceSavingsPercentageMargin savingsPercentage'})
brand_element = soup.find('span', attrs={'class':'a-size-base po-break-word'})
seller_element = soup.find('span',attrs={'class': 'a-size-small␣
 ↪mbcMerchantName'})
# Extract the text values from the elements
title = title_element.get_text().strip() if title_element else "Title not found"
price_str = price_element.get_text().replace(',', '') if price_element else␣
 ↪"Price not found"
price = float(price_str) if price_str else None
discount_str = discount_element.get_text().strip() if discount_element else "0%"
discount_str = discount_str.replace('-','')
discount = int(discount_str.replace('%', '')) if discount_str else 0
```

```
brand = brand_element.get_text().strip() if brand_element else "None"
seller = seller_element.get_text().strip() if seller_element else 'seller␣
  ↪details not found'
print(title)
print(price)
print(discount)
print(brand)
print(seller)
```

```
Apple iPhone 14 (128 GB) - Starlight
66999.0
16
Apple
Appario Retail Private Ltd
```

[3]:
```
import datetime
today = datetime.date.today()
print(today)
```

```
2023-06-30
```

[4]:
```
import csv
header = ['Date', 'Product', 'Price', 'Discount', 'Brand', 'Seller']
data =[today, title, price, discount, brand, seller]

with open("Amazonwebscrapperdataset.csv",'w',newline='', encoding='UTF8') as f:
    writer = csv.writer(f)
    writer.writerow(header)
    writer.writerow(data)
```

[5]:
```
import pandas as pd
df= pd.read_csv(r"Amazonwebscrapperdataset.csv")
```

[6]:
```
print(df)
```

```
         Date                               Product    Price  Discount  Brand  \
0  2023-06-30  Apple iPhone 14 (128 GB) - Starlight  66999.0        16  Apple

                       Seller
0  Appario Retail Private Ltd
```

[7]:
```
def check_price():

    URL = 'https://www.amazon.in/Apple-iPhone-14-128GB-Starlight/dp/B0BDK8LKPJ/
  ↪ref=sr_1_4?
  ↪crid=Y2Q6HTL0UEY3&keywords=iphone+13&qid=1687979356&sprefix=iphone+13%2Caps%2C284&sr=8-4'

    # Send a GET request to the URL
    response = requests.get(URL)

    # Parse the HTML content
    soup = BeautifulSoup(response.content, 'html.parser')

    # Find the title and price elements using appropriate selectors
    title_element = soup.find('span', attrs={'id': 'productTitle'})
    price_element = soup.find('span', attrs={'class': 'a-price-whole'})
    discount_element = soup.find('span', attrs={'class': 'a-size-large␣
↪a-color-price savingPriceOverride aok-align-center␣
↪reinventPriceSavingsPercentageMargin savingsPercentage'})
    brand_element = soup.find('span', attrs={'class':'a-size-base␣
↪po-break-word'})
    seller_element = soup.find('span',attrs={'class': 'a-size-small␣
↪mbcMerchantName'})
    # Extract the text values from the elements
    title = title_element.get_text().strip() if title_element else "Title not␣
↪found"
    price_str = price_element.get_text().replace(',', '') if price_element else␣
↪"Price not found"
    price = float(price_str) if price_str else None
    discount_str = discount_element.get_text().strip() if discount_element else␣
↪"0%"
    discount_str = discount_str.replace('-','')
    discount = int(discount_str.replace('%', '')) if discount_str else 0
    brand = brand_element.get_text().strip() if brand_element else "None"
    seller = seller_element.get_text().strip() if seller_element else 'seller␣
↪details not found'
    import datetime
    today = datetime.date.today()
    import csv
    header = ['Date', 'Product', 'Price', 'Discount', 'Brand', 'Seller']
    data =[today, title, price, discount, brand, seller]

    with open("Amazonwebscrapperdataset.csv",'a+',newline='', encoding='UTF8')␣
↪as f:
        writer = csv.writer(f)
        writer.writerow(data)

    if (price < 67000):
        send_mail()
```

[9]:

```
while(True):
    check_price()
    time.sleep(5)  # checks the price once every single day
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In[9], line 2
      1 while(True):
----> 2     check_price()
      3     time.sleep(5)  # checks the price once every single day

Cell In[7], line 37, in check_price()
     34     writer.writerow(data)
     36 if (price < 67000):
---> 37     send_mail()

Cell In[8], line 12, in send_mail()
      9 body = "N Sanjay Rao, This is the moment we have been waiting for. Now␣
  ↪is your chance to buy the phone of your dreams. Don't mess it up! Link here:␣
  ↪https://www.amazon.in/Apple-iPhone-14-128GB-Starlight/dp/B0BDK8LKPJ/ref=sr_1_1?
  ↪crid=Y2Q6HTL0UEY3&keywords=iphone+13&qid=1687979356&sprefix=iphone+13%2Caps%2C284&sr=8-4"
     10 message = f"Subject: {subject}\n\n{body}"
---> 12␣
  ↪server.sendmail('nandisanjay.ns@gmail.com', 'nadisanjay.ns@gmail.com', message)
     13 server.quit()

File ~\AppData\Local\Programs\Python\Python311\Lib\smtplib.py:892, in SMTP.
  ↪sendmail(self, from_addr, to_addrs, msg, mail_options, rcpt_options)
    890     to_addrs = [to_addrs]
    891 for each in to_addrs:
--> 892     (code, resp) = self.rcpt(each, rcpt_options)
    893     if (code != 250) and (code != 251):
    894         senderrs[each] = (code, resp)

File ~\AppData\Local\Programs\Python\Python311\Lib\smtplib.py:554, in SMTP.
  ↪rcpt(self, recip, options)
    552     optionlist = ' ' + ' '.join(options)
    553 self.putcmd("rcpt", "TO:%s%s" % (quoteaddr(recip), optionlist))
--> 554 return self.getreply()

File ~\AppData\Local\Programs\Python\Python311\Lib\smtplib.py:398, in SMTP.
  ↪getreply(self)
    396 while 1:
    397     try:
--> 398         line = self.file.readline(_MAXLINE + 1)
    399     except OSError as e:
    400         self.close()

File ~\AppData\Local\Programs\Python\Python311\Lib\socket.py:706, in SocketIO.
  ↪readinto(self, b)
    704 while True:
```

```
    705     try:
--> 706         return self._sock.recv_into(b)
    707     except timeout:
    708         self._timeout_occurred = True

File ~\AppData\Local\Programs\Python\Python311\Lib\ssl.py:1278, in SSLSocket.
 ↪recv_into(self, buffer, nbytes, flags)
   1274     if flags != 0:
   1275         raise ValueError(
   1276           "non-zero flags not allowed in calls to recv_into() on %s" %
   1277           self.__class__)
-> 1278     return self.read(nbytes, buffer)
   1279 else:
   1280     return super().recv_into(buffer, nbytes, flags)

File ~\AppData\Local\Programs\Python\Python311\Lib\ssl.py:1134, in SSLSocket.
 ↪read(self, len, buffer)
   1132 try:
   1133     if buffer is not None:
-> 1134         return self._sslobj.read(len, buffer)
   1135     else:
   1136         return self._sslobj.read(len)

KeyboardInterrupt:
```

[10]:

```python
import pandas as pd
df=pd.read_csv("Amazonwebscrapperdataset.csv")
print(df)
```

```
         Date                               Product    Price  Discount  Brand  \
0  2023-06-30  Apple iPhone 14 (128 GB) - Starlight  66999.0        16  Apple
1  2023-06-30  Apple iPhone 14 (128 GB) - Starlight  66999.0        16  Apple
2  2023-06-30  Apple iPhone 14 (128 GB) - Starlight  66999.0        16  Apple
3  2023-06-30  Apple iPhone 14 (128 GB) - Starlight  66999.0        16  Apple
4  2023-06-30  Apple iPhone 14 (128 GB) - Starlight  66999.0        16  Apple

                       Seller
0  Appario Retail Private Ltd
1  Appario Retail Private Ltd
2  Appario Retail Private Ltd
3  Appario Retail Private Ltd
4  Appario Retail Private Ltd
```

[8]:

```python
def send_mail():
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.ehlo()
    server.login('nandisanjay.ns@gmail.com', 'rbovgaxtsuxgrlto')

    subject = "The Phone you want is below Rs 67000 /-! Now is your chance to␣
 ↪buy!"
    body = "N Sanjay Rao, This is the moment we have been waiting for. Now is␣
 ↪your chance to buy the phone of your dreams. Don't mess it up! Link here:␣
 ↪https://www.amazon.in/Apple-iPhone-14-128GB-Starlight/dp/B0BDK8LKPJ/
 ↪ref=sr_1_4?
 ↪crid=Y2Q6HTL0UEY3&keywords=iphone+13&qid=1687979356&sprefix=iphone+13%2Caps%2C284&sr=8-4"
    message = f"Subject: {subject}\n\n{body}"

    server.sendmail('nandisanjay.ns@gmail.com', 'nadisanjay.ns@gmail.com',␣
 ↪message)
    server.quit()

# Rest of your code...
```