

Proiect Individual

EXTRAGEREA DATELOR DESPRE PRODUSELE UNUI MAGAZIN ONLINE FOLOSIND TEHNICA DE WEB SCRAPING

Nume: ASAVOAEI ANDREEA-DENISA
INFORMATICĂ ROMÂNĂ, ANUL 2, GRUPA 1

I. DESCRIERE PROIECT

Obiectivele

Programul implementat extrage date cu privire la ofertele de notebook-uri puse la dispoziție pe site-ul web al cunoscutului magazin Altex. Informațiile se referă la titlurile laptopurilor, prețurile actuale, prețurile inițiale – dacă au fost aplicate anumite reduceri cu mențiunile de rigoare –, stocurile produselor, link-urile către produse pentru informații suplimentare și scurte specificații despre acestea. Datele sunt stocate într-un fișier de tip .csv, asemănător cu un tabel, pentru parcurgerea mai facilă și mai rapidă a informațiilor.

Descrierea funcționalităților aplicației

Această aplicație este construită pe baza a trei funcții principale care reprezintă și cele mai importante etape ale procesului:

1. *Extragerea datelor de pe site-ul web* – realizată în funcția *mainFunction()* detaliată la punctul V, cu ajutorul librăriei *BeautifulSoup* oferită de Python;
2. *Construirea listei cu datele extrase* – realizată în funcția *construireListaProduse()* folosind ca structură de date lista de dicționare;
3. *Scrierea datelor din lista creată într-un fișier cu extensia .csv sub formă tabelară* – realizată în funcțiile *scriereInFisierCSV()* și *appendInFisierCSV()*.

Grupul țintă

Proiectul implementat are drept grup țintă indivizii din orice categorie de vârstă care au acces la un computer cu o conexiune la internet, care au minime cunoștințe de operare a computerului și care sunt interesați fie de ofertele de laptopuri puse la dispoziție de către magazinul online Altex, fie de piața de vânzări actuală, în scopuri personale.

Utilitatea și motivarea

Prin realizarea acestei aplicații s-a dorit căutarea mai rapidă a produselor disponibile pe site-ul antementionat prin renunțarea la utilizarea browser-ului care ar putea consuma mai multe resurse decât este de dorit. Se poate economisi o parte a memoriei RAM și chiar timp real deoarece timpul de execuție al programului este de câteva secunde, iar deschiderea unui browser și căutarea site-ului, respectiv căutarea categoriei de produse, poate varia în funcție de optimizarea browserului, calitatea componentelor hardware ale computerului de pe care se operează sau dexteritatea utilizatorului de la câteva zeci de secunde până la mai mult de un minut.

De asemenea, produsele sunt stocate într-un fișier cu extensia .csv care face parcurgerea mai rapidă și mai facilă pentru un utilizator interesat mai mult de prețuri și specificații decât de imagini ale produselor sau pentru un utilizator mai puțin experimentat.

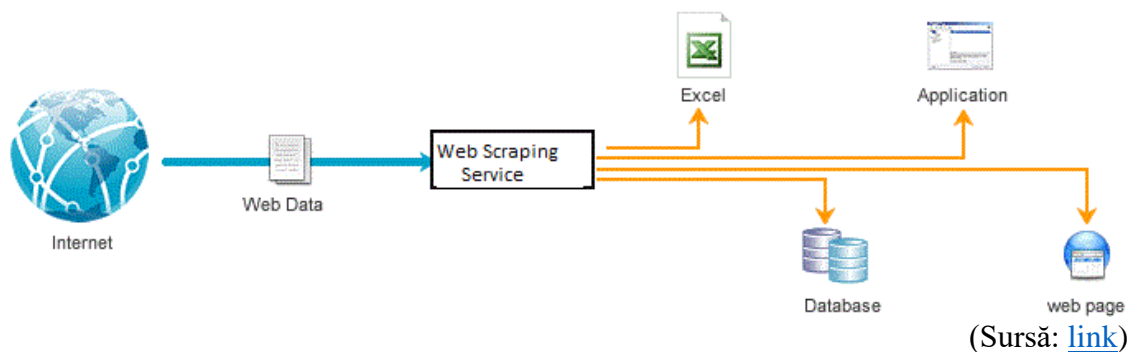
Alegerea acestui magazin online pe baza căruia să fie implementat programul are legătură strict cu felul în care este conceput site-ul, acesta având un cod HTML cu o structură organizată și relativ ușor de parcurs pentru conceperea aplicației.

II. DESCRIERE ARHITECTURĂ/DESIGN PROIECT

Input

Datele de intrare(input) sunt reprezentate de cele 15 pagini cu produse asociate site-ului web ale căror informații vor fi extrase.

Process



Programul realizat preia, una câte una, paginile web din input și caută informațiile dorite: denumirile, reducerile(dacă există), prețurile de dinainte și de după reducere, link-urile afiliate paginilor produselor și specificațiile din titluri. Aceste informații vor fi structurate în program sub forma unei liste cu dicționare. Fiecare dicționar corespunde fiecărui produs. Datele din lista populată vor fi salvate într-un fișier cu extensia .csv care poate fi accesat și filtrat rapid cu ajutorul aplicației Excel(Microsoft Office).

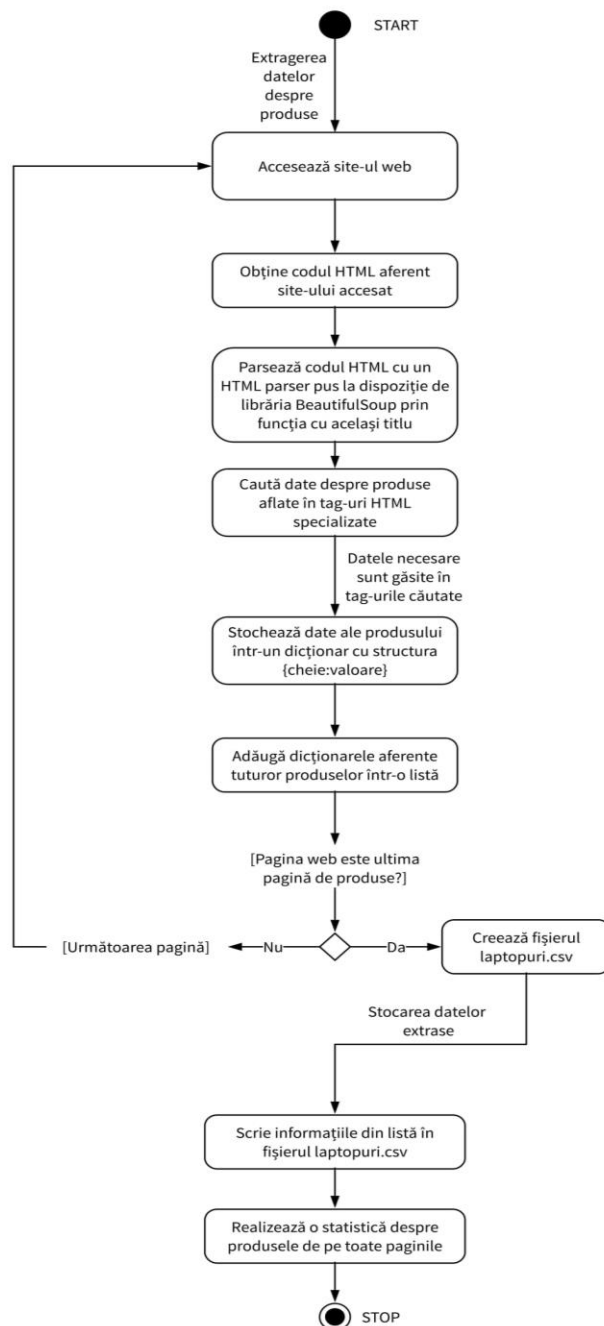
Output

Datele de ieșire(output) sunt reprezentate de fișierul cu extensia .csv conținând informații despre produse(titlu, are reducere(da/nu), reducere(procent/sumă redusă), preț actual, preț inițial, status, link, specificații) și de o scurtă statistică afișată pe consolă cu numărul notebook-urilor aflate la reducere, numărul notebook-urilor situate în anumite intervale de prețuri și cel mai vândut laptop de pe site.

Informații implementare

1. Arhitectură: desktop
2. Limbaj de programare: Python
3. IDE/Text Editor: Python IDE(Standard)/Sublime Text Editor

Diagramă activități UML



III. IMPLEMENTARE

Software suplimentar folosit

Pentru realizarea acestei aplicații s-a utilizat Python, un limbaj de programare dinamic, care pune accentul pe „curățenia” și simplitatea codului. Acest limbaj dispune

de o multitudine de biblioteci și module utile, printre cele utilizate regăsindu-se *urllib.request* și *beautifulsoup4*.

Modulul *urllib.request* definește clase și funcții care ajută la deschiderea URL-urilor într-un univers complex și necesită instalarea pachetului *Requests*.

Modulul *csv* implementează clase pentru citirea și scrierea datelor sub formă tabelară în format CSV(Comma Separated Value).

Biblioteca *beautifulsoup4* vine în ajutorul tehnicii de webscraping cu un parser HTML sau XML care analizează datele primite, de obicei de tip cod HTML. Tehnica de *webscraping* presupune extragerea unor date publice de pe anumite site-uri prin parsarea codului HTML asociat unei pagini. Aceste informații pot fi ulterior prelucrate cu diverse scopuri precum analizarea de date, stocarea acestora într-o bază de date, statistică sau altele.

Realizare prototip

Înainte de realizarea propriu-zisă a prototipului a fost necesară conceperea unei diagrame cu etapele prin care va trece programul odată executat, iar mai apoi o structură a fișierului în care se vor salva datele pentru a defini informațiile necesare de pe site-ul web. Din motive de utilitate, s-a dorit obținerea denumirilor produselor, reducerilor, prețurilor actuale și inițiale pentru comparație, scurte specificații pentru o viziune de ansamblu și un link în cazul în care utilizatorul va fi interesat de un anumit laptop și are nevoie de informații suplimentare precum recenzii, imagini cu produsul, descrierea amănunțită a produsului oferită de producător, specificații detaliate, verificarea stocului unui magazin fizic și mai ales pentru plasarea unei comenzi a produsului dorit.

Pentru realizarea prototipului s-a apelat întâi funcția de deschidere a unui obiect de rețea printr-un URL destinat citirii. După ce a fost citit, a fost apelată funcția *BeautifulSoup* cu HTML parser-ul pentru a prelua elementele site-ului. Pentru a rezolva problema găsirii unei modalități de stocare temporară în program a produselor obținute de pe site, a fost aleasă modalitatea listei cu elemente alte liste. Pe parcurs a devenit evident faptul că lista cu elemente dicționare este o idee mai bună în ceea ce privește organizarea. A susținut această idee și existența claselor cu funcții necesare stocării dicționarilor în fișier CSV, ceea ce a minimizat dificultatea scrierii informațiilor în acel fișier.

După ce informațiile au fost extrase de pe site, acestea au fost stocate în lista menționată anterior, iar mai apoi scrise în fișierul CSV.

Pentru a depăși nivelul de prototip, s-a dorit reluarea procesului pentru fiecare pagină de produse în parte. Acest fapt a completat aplicația și a eliminat criteriile de afișare pe site impuse de cei ce au implementat site-ul deoarece, în cazul implementării programului doar pentru prima pagină, ar fi fost obținute doar notebook-urile cele mai vândute(dacă după acest criteriu sunt afișate pe site), restul fiind excluse.

IV. TESTARE ȘI REMEDIERE BUG-URI

La realizarea programului au fost întâmpinate anumite probleme, cum ar fi:

1. la fiecare scriere a informațiilor despre produse în fișierul CSV, era adăugat header-ul conținând titlurile coloanelor, ceea ce ridica o problemă de estetică deoarece titlurile s-ar fi repetat de 15 ori(numărul de pagini cu produse existent în momentul realizării)

aplicației) – rezolvată prin adăugarea funcției *appendInFisierCSV()* care este apelată începând cu a doua pagină. Această funcție realizează doar scrierea valorilor din listă, excluzând header-ul.

2. deoarece numărul de pagini poate varia în funcție de numărul de produse puse la dispoziție pe site, este de dorit a se contoriza numărul de pagini curent prin extragerea link-urilor aferente fiecărei pagini – nerezolvată, deoarece clasa din tag-ul în care se regăsesc informațiile necesare este utilizată de mai multe ori în codul HTML, ceea ce creează confuzie la apelul funcției *findAll()* sau *find()*.
3. în funcțiile *produseIntervalLei()* și *pretMaximLista()* se prelucrează prețul produselor, acesta fiind un șir de caractere. Compararea unui șir de caractere cu o valoare numerică precum capetele intervalului transmis ca parametru funcției *produseIntervalLei()* nu este posibilă – rezolvată prin convertirea prețului la tipul de date *float*.
4. deoarece există mai multe pagini cu produse, vor exista și mai multe liste create. Așadar, sunt necesare foarte multe apeluri ale funcțiilor *produseReduse()* și *produseIntervalLei()* – rezolvată prin concatenarea tuturor listelor rezultând o listă de dicționare cu toate produsele.
5. pentru obținerea titlurilor laptopurilor, s-a găsit șablonul "Laptop [BRAND] [MODEL], [SPECIFICAȚII]" în titlurile afișate pe site și, conform aceluși șablon, se poate extrage denumirea produsului până la prima virgulă. Totuși, există anumite produse cum este *Pachet Laptop MYRIA MY8311PK cu Windows 10 Home preinstalat + Office 365 + Husa + Mouse* care nu urmăresc șablonul și ridică probleme în această privință deoarece câmpul *titlu* din dicționar nu poate fi completat – rezolvată temporar prin excluderea acestui tip de produse până la găsirea unei alternative pentru șablonul respectiv.

V. DOCUMENTARE

Documentare cod

Acest program conține un singur fișier cu extensia *.py* în care se află codul sursă.

În primă instanță, sunt importate anumite funcții din modulele antemenționate. Din *urllib.request* se importă *urlopen*, funcție cu ajutorul căreia este accesat URL-ul site-

ului dorit transmis ca parametru de tip șir de caracter sau obiect de tipul *Request*. Din biblioteca *Standard* se importă *csv*, modul cu care pot fi citite și scrise date într-un fișier de tip CSV. Din *bs4* este importată funcția *BeautifulSoup* cu care este transformată pagina web transmisă ca parametru alături de un *HTML/XML parser* în format *BeautifulSoup* pentru a putea prelucra datele.

```
projectPl.py
1 from urllib.request import urlopen
2 import csv
3 from bs4 import BeautifulSoup
```

Prima funcție definită este *construireListaProduse()* care are ca parametru site-ul web transformat în format *BeautifulSoup* utilizând HTML parser-ul datorită formatului pagini. Se inițializează o listă vidă care va conține dicționare pentru fiecare laptop găsit pe site. Dicționarele sunt de forma {cheie1:valoare1, cheie2:valoare2,...,cheieN:valoareN} și, în această situație, cheie1=*Titlu*, cheie2=*Are reducere*, cheie3=*Reducere*, cheie4=*Preț actual*, cheie5=*Preț vechi*, cheie6=*Status produs*, cheie7=*Link produs*, cheie8=*Specificații produs*.

Prima instrucțiune *for* caută cu ajutorul funcției *findAll()* în site-ul transmis ca parametru toate tag-urile *div* cu clasa *Product*, acestea reprezentând informațiile despre produsele oferite de site. Este returnată o listă în care, mai apoi, se caută cu ajutorul aceleiași instrucțiuni tag-ul *a* cu clasa *Product-name*. Acest tag va conține titlurile produselor. Se inițializează dicționarul curent, apoi se obține din iteratorul instrucțiunii *for* titlul obiectului curent prin apelarea funcției *get()* cu parametrul *title* care reprezintă atributul tag-ului *a*. Pentru a prelua din titlul doar brand-ul și seria laptopului, se caută în titlu prima virgulă care separă denumirea produsului de specificații. Dacă aceasta nu este găsită, se trece la următorul produs, excluzându-l pe cel care nu urmărește șablonul. În caz contrar, cheii *Titlu* i se va atribui valoarea șirului de caractere extras.

În următoarea instrucțiune *for* sunt căutate în mod identic produsele incluse în tag-ul *div* cu clasa *Badge*. Mai apoi, pentru a verifica dacă produsele au reducere, se apelează funcția *find()* pentru argumentele *div* cu clasa *Badge-reducere*. Dacă laptopul curent nu dispune de reducere, cheii *Are reducere* i se atribuie valoarea *NU*, iar în câmpul *Reducere* este lăsat spațiu liber. În caz contrar, cheii *Are reducere* i se atribuie valoarea *DA*, iar cheii *Reducere* i se va atribui valoarea numerică din textul extras din acel tag *div*. La ieșirea din acest ciclu *for*, se verifică dacă au rămas produse neavând tag-ul *div* cu clasa *Badge*, iar celor identificate le este atribuită cheilor *Are reducere* valoarea *NU*.

Pentru a identifica prețurile actuale ale laptopurilor de pe site, se caută cu ajutorul funcției *find()* elementele cu atributul *itemprop* având valoarea *price*. De aici se extrage valoarea din atributul *content* cu ajutorul funcției *get()*. Această valoare i se atribuie cheii *Preț actual*, adăugând și moneda în care sunt afișate prețurile, adică *lei*.

Pentru identificarea prețurilor vechi/inițiale, se caută tag-ul *div* cu clasa *Price-old*. Dacă un produs nu este la reducere, nu va avea preț vechi și, prin urmare, este lăsat spațiu liber. În caz contrar, se extrage valoarea și i se atribuie cheii *Preț vechi* adăugând moneda precum în situația anterioară.

Stocul unui produs este găsit în mod similar, căutând informațiile oferite de tag-ul *div* cu clasa *Status*. Valoarea text găsită în acel tag este transmisă cheii *Status produs*.

Link-ul către pagina ce conține produsul de pe site pentru informații suplimentare este extras cu ajutorul funcției *findAll()* având ca argumente tag-ul *a* cu clasa *Product-photoTrigger js-ProductClickListener* și cu ajutorul funcției *get()* cu argumentul *href*, atributul care reține link-ul asociat. Acest link este atribuit cheii *Link produs* din dicționar.

Pentru specificațiile laptopurilor, se caută același tag de mai sus și se obțin specificațiile din atributul *title*, acestea urmând să fie asociate cu cheia *Specificații produs*.

La final, fiecare dicționar care corespunde fiecărui produs de pe pagina web va fi adăugat listei inițializate la începutul funcției și se va returna lista completă.

```

4 def construireListaProduce(soupSite):
5     productList=[]
6     for containerProduct in soupSite.findAll("div",{"class":"Product"}):
7         for containerTitle in containerProduct.findAll("a",{"class":"Product-name"}):
8             productDict={}
9             s=containerTitle.get("title")
10            pos=s.find(",")
11            if(pos<0):
12                continue
13            productDict["Titlu"]=s[0:pos]
14            for containerSale in containerProduct.findAll("div",{"class":"Badge"}):
15                productDiscount=containerSale.find("div",class_="Badge-reducere")
16                if(productDiscount==None):
17                    productDict["Are reducere"]="NU"
18                    productDict["Reducere"]=" "
19                else:
20                    productDict["Are reducere"]="DA"
21                    productDict["Reducere"]=productDiscount.get_text()[8:]
22            if(containerProduct.find("div",class_="Badge")==None):
23                productDict["Are reducere"]="NU"
24            currentPrice=containerProduct.find(itemprop="price")
25            s=currentPrice.get("content")
26            productDict["Pret actual"]=s+" lei"
27            productOldPrice=containerProduct.find("div",class_="Price-old")
28            if(productOldPrice==None):
29                productDict["Pret vechi"]=" "
30            else:
31                productDict["Pret vechi"]=productOldPrice.get_text()+" lei"
32            productStock=containerProduct.find("div",class_="Status")
33            productDict["Status produs"]=productStock.get_text()
34            for containerLink in containerProduct.findAll("a",{"class":"Product-photoTrigger js-ProductClickListener"}):
35                s=containerLink.get("href")
36                productDict["Link produs"]=s
37            for containerSpecs in containerProduct.findAll("a",{"class":"Product-photoTrigger js-ProductClickListener"}):
38                s=containerSpecs.get("title")
39                pos=s.find(", ")
40                pos=pos+1
41                productDict["Specificatii produs"]="Procesor"+s[pos:]
42            productList.append(productDict)
43    return productList

```

Funcția *scriereInFisierCSV()* primește ca parametru lista de dicționare cu produse returnată anterior. Cu ajutorul variabilei *keys* se obțin denumirile cheilor din listă pentru a defini titlurile coloanelor din tabel. Folosind funcția *open()* se va deschide un fișier, *laptopuri.csv* (dacă acesta nu există, se va crea unul cu titlul respectiv), în modul *write* pentru a putea scrie date în acesta. Se folosește variabila *outputFile* pentru a reține forma finală a fișierului. În modulul *csv* există clasa *DictWriter* care permite maparea dicționarului cu fișierul *.csv*. Acesta preia cheile din dicționar, le stochează în variabila *dict_writer*, apoi, folosind funcția *writeheader()*, atașează valorile din *dict_writer* denumirilor coloanelor din tabel. În final, prin funcția *writerows()* se inserează înregistrările (valorile din dicționare) în fișier.

Funcția *appendInFisierCSV()* este similară, cu mențiunea că fișierul se deschide în modul *append* pentru a adăuga la finalul fișierului înregistrările, iar denumirile coloanelor nu se mai inserează deoarece există deja la început.

Prin funcția *produseReduce()* se caută în lista de dicționare de produse, laptopurile care au atribuită cheii *Are reducere* valoarea *DA* și se afișează numărul de astfel de produse.

Funcția *produseIntervalLei()* primește ca parametri lista de dicționare și capetele intervalului din care se dorește a fi prețul produselor. Se parcurge lista, se obține valoarea numerică asociată cheii *Preț actual*, se verifică dacă acea valoare se află în intervalul cerut și se contorizează cazurile care verifică condiția. La final se returnează numărul de produse.

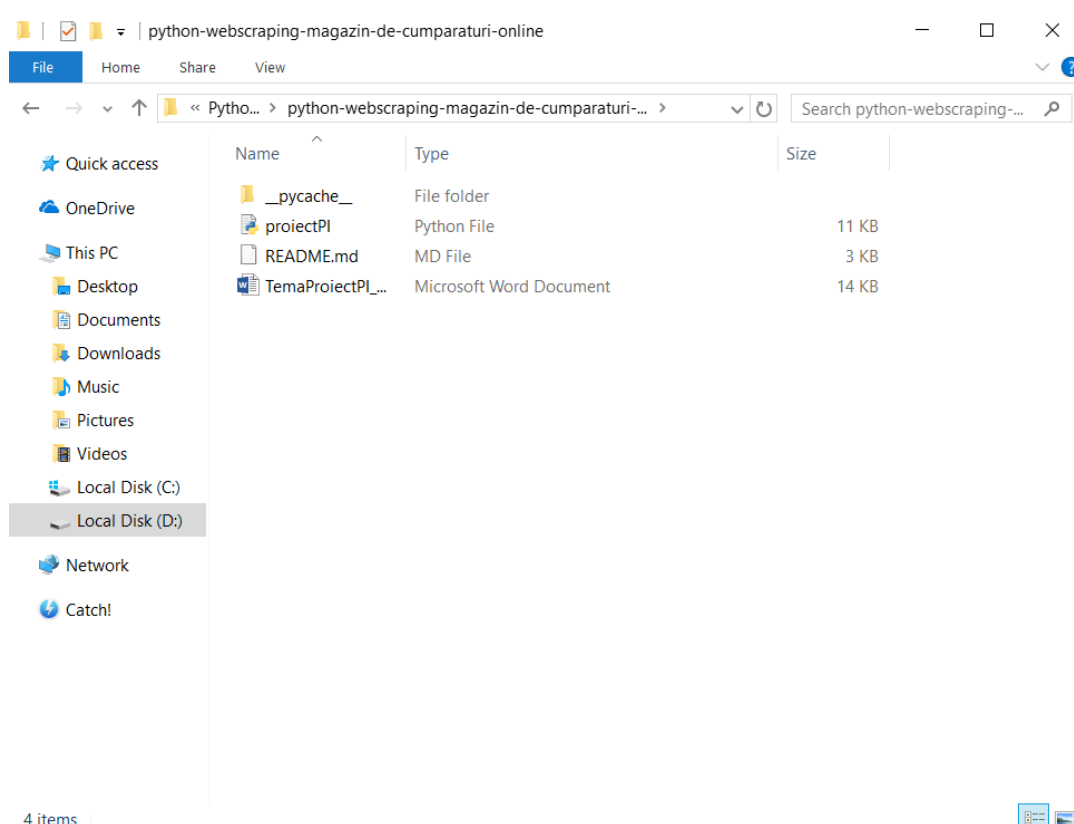
Pentru organizarea apelurilor funcțiilor definite anterior s-a folosit funcția *mainFunction()*.

Ghid de utilizare

Mențiuni:

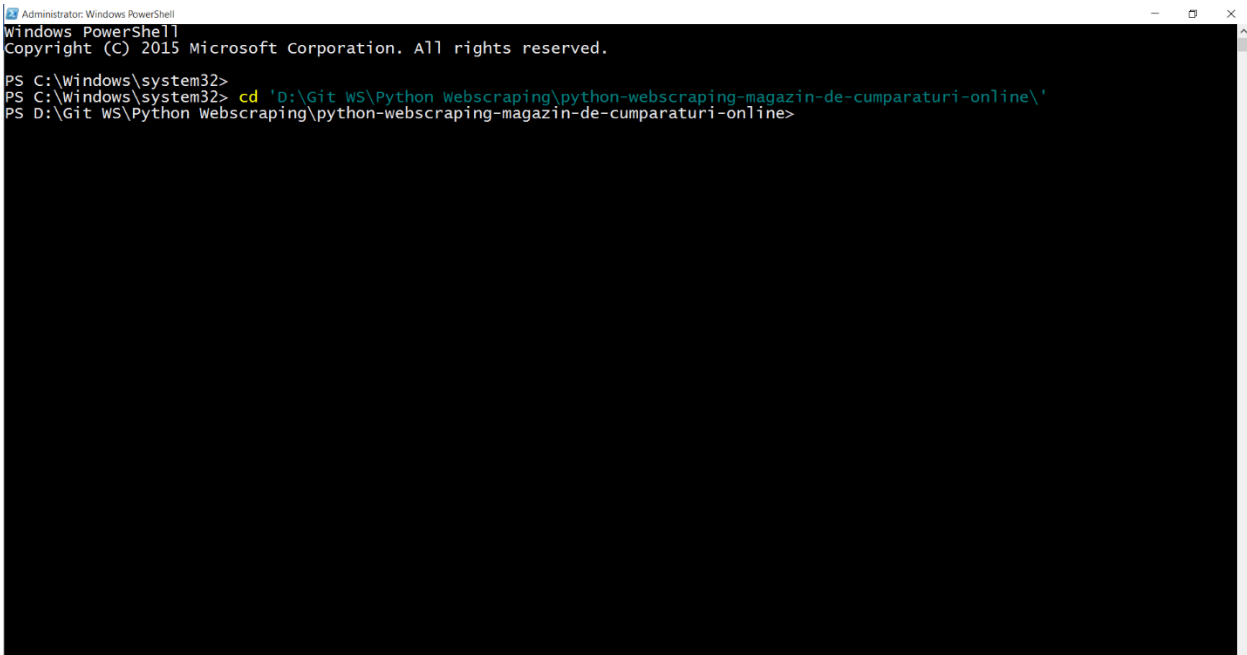
1. Pentru a putea rula cu succes programul, este necesară instalarea următoarelor echipamente software gratuite: Python 3.x și Beautiful Soup 4.
2. Acest ghid a fost realizat în data de 02.12.2018, deci este posibil ca informațiile afișate în el să nu mai fie concludente deoarece site-ul web poate fi actualizat oricând și informațiile despre produse pot fi modificate.

În captura de ecran de mai jos se poate observa conținutul inițial al directorului în care se află programul.



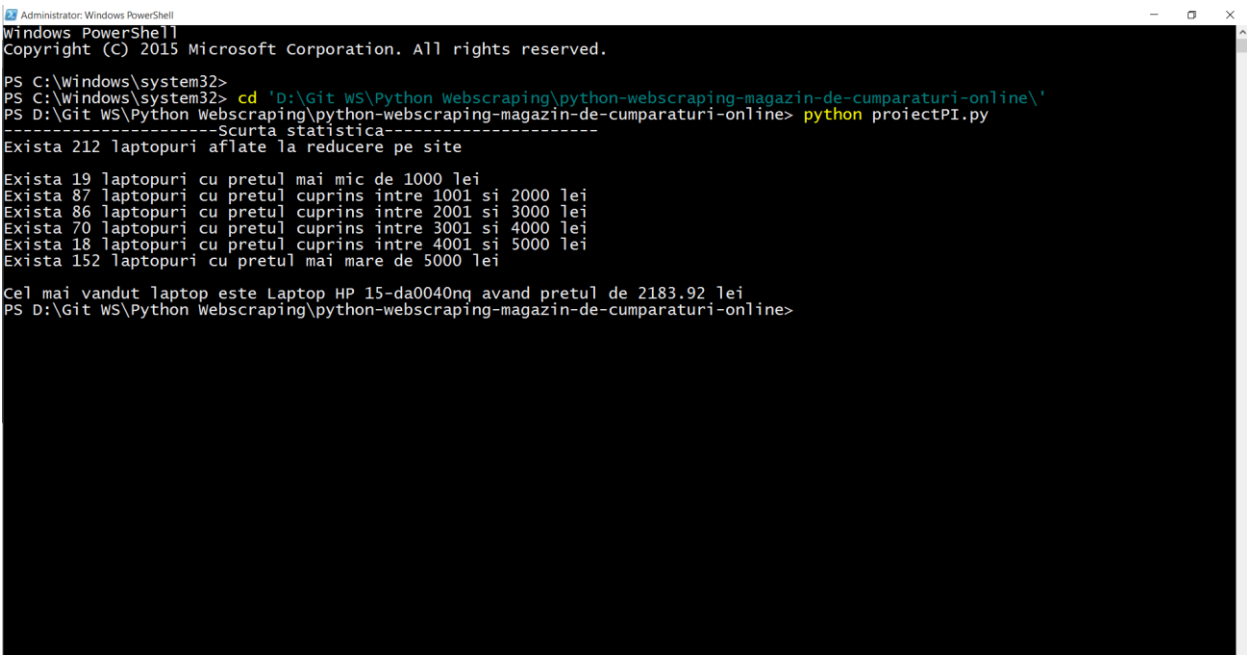
1. Executare din terminal

Pentru a rula fișierul *proiectPI.py*, se deschide un terminal în modul administrator. Cu ajutorul comenzii *cd + cale_fișier* se accesează folderul în care se află proiectul, iar cu ajutorul comenzii *python* poate fi rulată aplicația, astfel:



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32>
PS C:\Windows\system32> cd 'D:\Git WS\Python Webscraping\python-webscraping-magazin-de-cumparaturi-online\'
PS D:\Git WS\Python Webscraping\python-webscraping-magazin-de-cumparaturi-online>
```



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

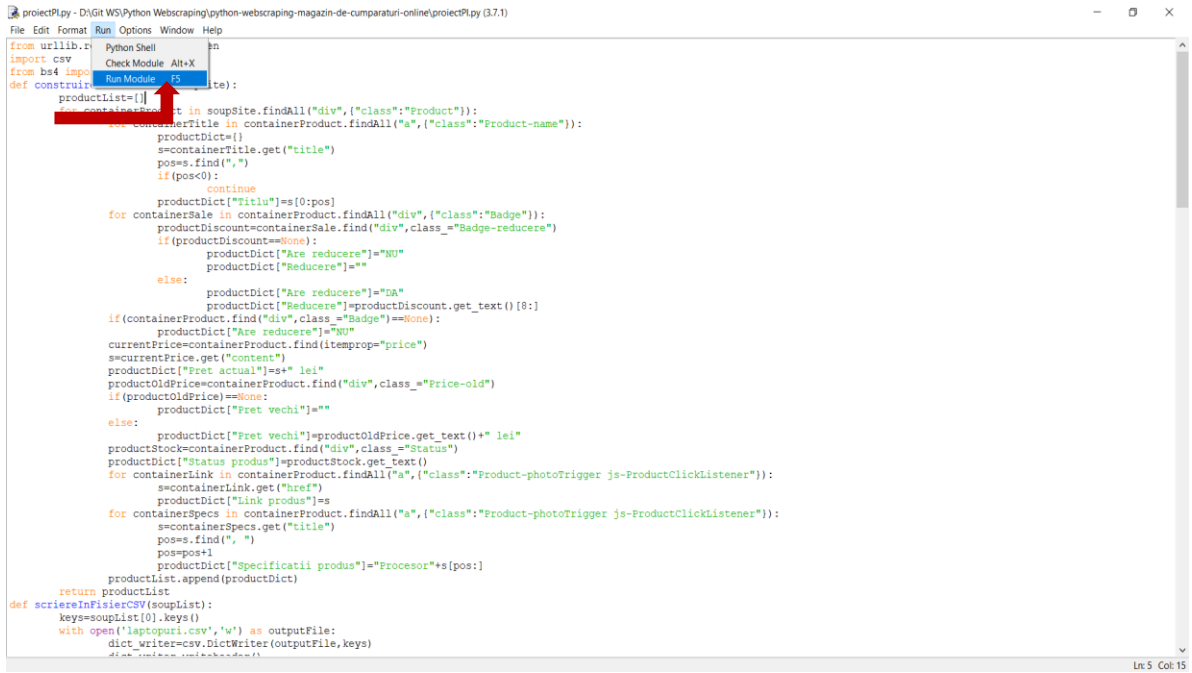
PS C:\Windows\system32>
PS C:\Windows\system32> cd 'D:\Git WS\Python Webscraping\python-webscraping-magazin-de-cumparaturi-online\'
PS D:\Git WS\Python Webscraping\python-webscraping-magazin-de-cumparaturi-online> python proiectPI.py
-----Scurta statistica-----
Exista 212 laptopuri aflate la reducere pe site

Exista 19 laptopuri cu pretul mai mic de 1000 lei
Exista 87 laptopuri cu pretul cuprins intre 1001 si 2000 lei
Exista 86 laptopuri cu pretul cuprins intre 2001 si 3000 lei
Exista 70 laptopuri cu pretul cuprins intre 3001 si 4000 lei
Exista 18 laptopuri cu pretul cuprins intre 4001 si 5000 lei
Exista 152 laptopuri cu pretul mai mare de 5000 lei

Cel mai vandut laptop este Laptop HP 15-da0040ng avand pretul de 2183.92 lei
PS D:\Git WS\Python Webscraping\python-webscraping-magazin-de-cumparaturi-online>
```

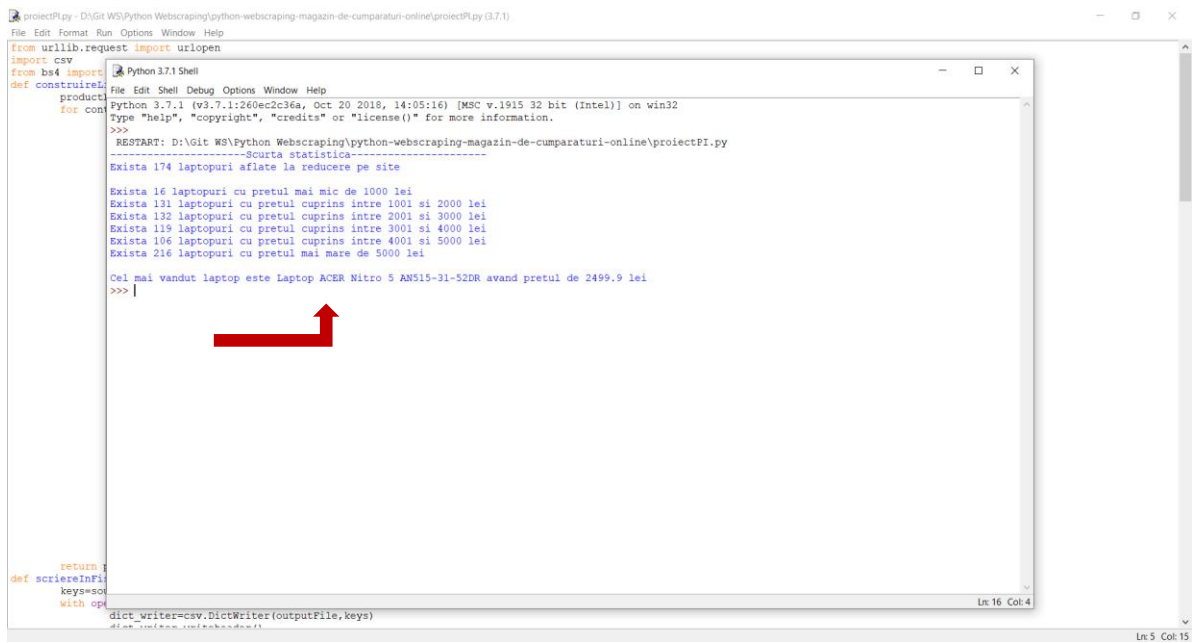
În fereastra aceasta se poate observa o scurtă statistică a produselor.

Executare din Python Standard IDE



```
projectPI.py - D:\Git WS\Python Webscraping\python-webscraping-magazin-de-cumparaturi-online\projectPI.py (3.7.1)
File Edit Format Run Options Window Help
from urllib.request import urlopen
import csv
from bs4 import BeautifulSoup
def construiesteProdus(site):
    productList=[]
    for containerProduct in soupSite.findAll("div",{"class":"Product"}):
        containerTitle = containerProduct.findAll("a",{"class":"Product-name"}):
        productDict={}
        s=containerTitle.get("title")
        pos=0
        if(pos<0):
            continue
        productDict["Title"]=s[pos:]
        for containerSale in containerProduct.findAll("div",{"class":"Badge"}):
            productDiscount=containerSale.find("div",class_="Badge-reducere")
            if(productDiscount==None):
                productDict["Are reducere"]="NU"
                productDict["Reducere"]=" "
            else:
                productDict["Are reducere"]="DA"
                productDict["Reducere"]=productDiscount.get_text()[8:]
        if(containerProduct.find("div",class_="Badge")==None):
            productDict["Are reducere"]="NU"
            currentPrice=containerProduct.find(itemprop="price")
            s=currentPrice.get("content")
            productDict["Pret actual"]=s+" lei"
            productOldPrice=containerProduct.find("div",class_="Price-old")
            if(productOldPrice==None):
                productDict["Pret vechi"]=" "
            else:
                productDict["Pret vechi"]=productOldPrice.get_text()+" lei"
        productStock=containerProduct.find("div",class_="Status")
        productDict["Status produs"]=productStock.get_text()
        for containerLink in containerProduct.findAll("a",{"class":"Product-photoTrigger js-ProductClickListener"}):
            s=containerLink.get("href")
            productDict["Link produs"]=s
        for containerSpecs in containerProduct.findAll("a",{"class":"Product-photoTrigger js-ProductClickListener"}):
            s=containerSpecs.get("title")
            pos=s.find(" ", " ")
            pos=pos+1
            productDict["Specificatii produs"]="Procesor"+s[pos:]
        productList.append(productDict)
    return productList
def scriereInFisierCSV(soupList):
    keys=soupList[0].keys()
    with open('laptopuri.csv','w') as outputFile:
        dict_writer=csv.DictWriter(outputFile,keys)
        dict_writer.writerow(keys)
```

Ln 5 Col 15



```
projectPI.py - D:\Git WS\Python Webscraping\python-webscraping-magazin-de-cumparaturi-online\projectPI.py (3.7.1)
File Edit Format Run Options Window Help
from urllib.request import urlopen
import csv
from bs4 import BeautifulSoup
def construiesteProdus(site):
    productList=[]
    for containerProduct in soupSite.findAll("div",{"class":"Product"}):
        containerTitle = containerProduct.findAll("a",{"class":"Product-name"}):
        productDict={}
        s=containerTitle.get("title")
        pos=0
        if(pos<0):
            continue
        productDict["Title"]=s[pos:]
        for containerSale in containerProduct.findAll("div",{"class":"Badge"}):
            productDiscount=containerSale.find("div",class_="Badge-reducere")
            if(productDiscount==None):
                productDict["Are reducere"]="NU"
                productDict["Reducere"]=" "
            else:
                productDict["Are reducere"]="DA"
                productDict["Reducere"]=productDiscount.get_text()[8:]
        if(containerProduct.find("div",class_="Badge")==None):
            productDict["Are reducere"]="NU"
            currentPrice=containerProduct.find(itemprop="price")
            s=currentPrice.get("content")
            productDict["Pret actual"]=s+" lei"
            productOldPrice=containerProduct.find("div",class_="Price-old")
            if(productOldPrice==None):
                productDict["Pret vechi"]=" "
            else:
                productDict["Pret vechi"]=productOldPrice.get_text()+" lei"
        productStock=containerProduct.find("div",class_="Status")
        productDict["Status produs"]=productStock.get_text()
        for containerLink in containerProduct.findAll("a",{"class":"Product-photoTrigger js-ProductClickListener"}):
            s=containerLink.get("href")
            productDict["Link produs"]=s
        for containerSpecs in containerProduct.findAll("a",{"class":"Product-photoTrigger js-ProductClickListener"}):
            s=containerSpecs.get("title")
            pos=s.find(" ", " ")
            pos=pos+1
            productDict["Specificatii produs"]="Procesor"+s[pos:]
        productList.append(productDict)
    return productList
def scriereInFisierCSV(soupList):
    keys=soupList[0].keys()
    with open('laptopuri.csv','w') as outputFile:
        dict_writer=csv.DictWriter(outputFile,keys)
        dict_writer.writerow(keys)
```

Ln 16 Col 4

Python 3.7.1 Shell

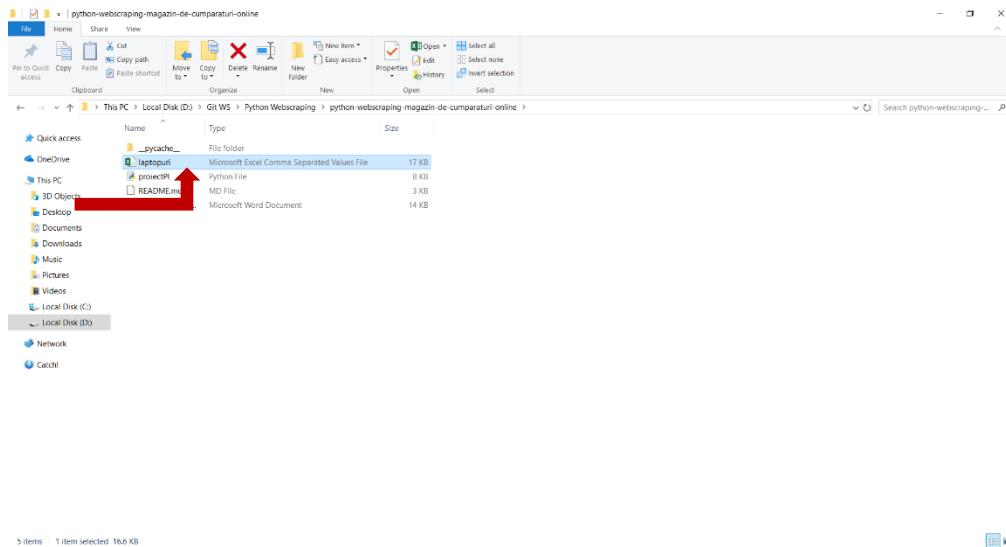
```
Python 3.7.1 (tags/v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\Git WS\Python Webscraping\python-webscraping-magazin-de-cumparaturi-online\projectPI.py
-----Scurta statistica-----
Exista 174 laptopuri afisate la reducere pe site

Exista 16 laptopuri cu pretul mai mic de 1000 lei
Exista 131 laptopuri cu pretul cuprins intre 1001 si 2000 lei
Exista 132 laptopuri cu pretul cuprins intre 2001 si 3000 lei
Exista 119 laptopuri cu pretul cuprins intre 3001 si 4000 lei
Exista 106 laptopuri cu pretul cuprins intre 4001 si 5000 lei
Exista 216 laptopuri cu pretul mai mare de 5000 lei

Cel mai vandut laptop este Laptop ACER Nitro 5 AN515-31-52DR avand pretul de 2499.9 lei
>>>
```

Ln 5 Col 15

2. Se deschide folderul în care se află proiectul și se poate observa apariția unui nou fișier, *laptopuri.csv*, indicat cu săgeată roșie. Acesta poate fi accesat cu ajutorul unui program precum Excel din suita Microsoft Office.



3. După accesarea fișierului, pentru a testa și compara rezultatele, se poate deschide în browser site-ul magazinului <https://altex.ro/laptopuri/cpl/>, observând că informațiile corespund. Săgețile roșii indică datele care trebuie comparate pentru a sesiza corectitudinea programului. După același șablon pot fi comparate restul produselor listate.

laptopuri - Excel															
Titlu															
1	Titlu	Are reduce	Reducere	Pret actual	Pret vechi	Status prov	Link produ	Specificatii produs							
2	Laptop ACER Nitro 5 AN515-31-52DR	DA	300 lei	2499.9 lei	2.799,90 lei	In Stoc	https://alt	Procesor Intel® Core™ i5-8250U pana la 3.4GHz, 15.6" Full HD, 8GB, 1TB, NVIDIA GeForce MX150 2GB, Linux							
3	Laptop Gaming ASUS UF FX705GM-EW019	DA	600 lei	5099.9 lei	5.699,90 lei	In Stoc	https://alt	Procesor Intel® Core™ i7-8750H pana la 4.1GHz, 17.3" Full HD, 8GB, 1TB + SSD 128GB, NVIDIA GeForce GTX 1060 6GB, Free Dos							
4	Laptop MYRIA MY8306	NU		599.9 lei		In Stoc	https://alt	Procesor Intel® Celeron® N3350 pana la 2.4GHz, 14.1" IPS HD, 4GB, eMMC 32GB, Intel® HD Graphics 500, Windows 10 Home							
5	Laptop LENOVO IdeaPad 530S-14ARR	DA	250 lei	2399.9 lei	2.649,90 lei	In Stoc	https://alt	Procesor AMD Ryzen 5 2500U pana la 3.6GHz, 14" Full HD, 8GB, SSD 256GB, AMD Radeon Vega 8, Free Dos							
6	Laptop 2 in 1 MYRIA MY8307	NU		499.9 lei		In Stoc	https://alt	Procesor Intel Atom Z8350 pana la 1.92GHz, 10.1" Touch, 2GB, eMMC 32GB, Intel® HD Graphics, Windows 10 Home							
7	Laptop Gaming ASUS FX503VD-E4082	DA	400 lei	2999.9 lei	3.399,90 lei	In Stoc	https://alt	Procesor Intel® Core™ i5-7300HQ pana la 3.5GHz, 15.6" Full HD, 8GB, 1TB + 8GB cache, NVIDIA GeForce GTX 1050 4GB, Free Dos							
8	Laptop ASUS ZenBook UX430UA-GV183T	DA	300 lei	2699.9 lei	2.999,90 lei	In Stoc	https://alt	Procesor Intel® Core™ i5-7200U pana la 3.1GHz, 14" Full HD, 4GB, SSD 256GB, Intel® HD Graphics 620, Windows 10							
9	Laptop 2 in 1 ASUS VivoBook Flip TP142UA-EC056	DA	400 lei	2599.9 lei	2.999,90 lei	In Stoc	https://alt	Procesor 14" Full HD Touch, Intel Core i3-7020U 2.3GHz, 4GB, SSD 256GB, Intel HD Graphics 620, Windows 10 Home							
10	Laptop HP 15-db0011nq	DA	350 lei	2349.9 lei	2.699,90 lei	In Stoc	https://alt	Procesor AMD Ryzen 5 2500U pana la 3.6GHz, 15.6" Full HD, 8GB, 1TB + SSD 128GB, AMD Radeon Vega 8, Free Dos							
11	Laptop LENOVO IdeaPad 320-17IKB	NU		2399.9 lei		In Stoc	https://alt	Procesor Intel® Core™ i5-7200U Processor (pana la 3.1GHz), 17.3", 8GB, 1TB, Intel® HD Graphics 620, Free Dos							
12	Laptop ASUS X542UA-DM521	DA	600 lei	1899.9 lei	2.499,90 lei	In Stoc	https://alt	Procesor Intel® Core™ i5-8250U pana la 3.4GHz, 15.6" Full HD, 4GB, 1TB, Intel® UHD Graphics 620, Endless							
13	Laptop LENOVO IdeaPad 330S-14IKB	DA	100 lei	1699.99 lei	1.799,99 lei	In Stoc	https://alt	Procesor Intel Core i3-8130U pana la 3.4GHz, 14" Full HD, 4GB, 2TB, Intel UHD Graphics 620, Free Dos, Platinum Grey							
14	Laptop HP 15-da0139nq	DA	400 lei	1599.9 lei	1.999,90 lei	In Stoc	https://alt	Procesor Intel Core i3-7100U 2.4GHz, 15.6" Full HD, 4GB, 1TB, Intel HD Graphics 620, Free Dos							
15	Laptop Lenovo IdeaPad 320-15ISK	DA	300 lei	1399.9 lei	1.699,90 lei	In Stoc	https://alt	Procesor Intel® Core™ i3-6006U 2.0GHz, 15.6", 4GB, 1TB, Intel® HD Graphics 520, Free Dos							

Laptopuri la cel mai mic pret din

https://altex.ro/laptopuri/cpl/

Apps

★ Bookmarks

Click aici catre cele mai bune promotii.

Cum cumpar?

Support Clienti

2x Diferenta

Magazine

Contact

021 9196

Inregistrare

Intra in cont

ALTEx

CEL MAI MIC PRET DIN ROMANIA*

Cauta produsul dorit

Contul meu

Cosul meu

Produse

Catalog oferte

Servicii

Finantare

Carriere

Acasa / Laptop - Desktop - IT / Laptopuri

General

Promotii (174)

In stoc (530)

Noutati (43)

Precomanda (5)

Brand

ASUS (212)

HP (152)

LENOVO (112)

DELL (108)

ACER (57)

APPLE (52)

MSI (21)

Mai mult

Categorii

Laptop 2 in 1 (107)

Laptop Business (24)

Laptop Gaming (152)

Laptop Home (318)

Laptop Ultraportabil (136)

2 in 1

Convertibil (94)

Detasabil (9)

Pret

Sub 500 (1)

500 - 1000 (16)

1000 - 1500 (69)

1500 - 2000 (69)

2000 - 3000 (132)

3000 - 4000 (119)

4000 - 5000 (106)

Mai mult

Tip

Starter (71)

Practic (170)

Performant (371)

Extrem (14)

Sistem de operare

Windows 10 S (5)

Windows 10 Home (241)

Windows 10 Pro (64)

macOS Sierra (26)

macOS Mojave (16)

Linux (42)

Ubuntu (49)

Mai mult

Dimensiune ecran (inch)

12.2 (1)

VÂNĂTOAREA ÎNCEPE

PENTRU O PERIOADĂ LIMITATĂ, CUMPĂRĂ O PLACĂ VIDEO GEFORCE GTX ELIGIBILĂ ȘI OBTINE MONSTER HUNTER: WORLD*

Cumpără acum

MONSTER HUNTER

Laptopuri

Sortează după: Cele mai vandute

Vezi: Tabel

Arată: 48

Pagina: 1 / 16

REDUCERE

300 lei

+ CADOU Intel Software pack

2.499,90

IN STOC

Laptop ACER Nitro 5 AN515-31-52DR, Intel® Core™ i5-8250U până la 3.4GHz, 15.6"

Adauga in cos

Verifica stoc magazin

Adauga in wishlist

Compara

REDUCERE

600 lei

+ CADOU

5.099,90

IN STOC

Laptop Gaming ASUS TUF FX705GM-EW019, Intel® Core™ i7-8750H până la

Adauga in cos

Verifica stoc magazin

Adauga in wishlist

Compara

599,90

IN STOC

Laptop MYRIA MY8306, Intel® Celeron® N3350 până la 2.4GHz, 14.1" IPS HD, 4GB,

Adauga in cos

Verifica stoc magazin

Adauga in wishlist

Compara

REDUCERE

250 lei

2.399,90

IN STOC

Laptop LENOVO IdeaPad 530S-14ARR, AMD Ryzen 5 2500U până la 3.6GHz, 14" Full HD,

Adauga in cos

Verifica stoc magazin

Adauga in wishlist

Compara

499,90

IN STOC

Laptop 2 in 1 MYRIA MY8307, Intel Atom Z8350 până la 1.92GHz, 10.1" Touch, 2GB,

Adauga in cos

Verifica stoc magazin

Adauga in wishlist

Compara

REDUCERE

400 lei

+ CADOU Intel Software pack

2.999,90

IN STOC

Laptop Gaming ASUS FX503VD-E4082, Intel® Core™ i5-7300HQ până la 3.5GHz, 15.6"

Adauga in cos

Verifica stoc magazin

Adauga in wishlist

Compara

REDUCERE

300 lei

+ CADOU Intel Software pack

2.699,90

IN STOC

Laptop ASUS ZenBook UX430UA-GV183T, Intel® Core™ i5-7200U până la

Adauga in cos

Verifica stoc magazin

Adauga in wishlist

Compara

REDUCERE

400 lei

2.599,90

IN STOC

Laptop 2 in 1 ASUS VivoBook Flip TP412UA-EC056T, 14" Full HD Touch, Intel Core i3-7020U

Adauga in cos

Verifica stoc magazin

Adauga in wishlist

Compara

REDUCERE

350 lei

2.349,90

IN STOC

Laptop HP 15-db0011mq, AMD Ryzen 5 2500U până la 3.6GHz, 15.6" Full HD, 8GB,

Adauga in cos

Verifica stoc magazin

Adauga in wishlist

Compara

REDUCERE

600 lei

+ CADOU Intel Software pack

2.399,90

IN STOC

Laptop LENOVO IdeaPad 320-17IKB, Intel® Core™ i5-7200U Processor (până la 3.1GHz),

Adauga in cos

Verifica stoc magazin

Adauga in wishlist

Compara

REDUCERE

600 lei

+ CADOU Intel Software pack

1.899,90

IN STOC

Laptop ASUS X542UA-DM521, Intel® Core™ i5-8250U până la 3.4GHz, 15.6" Full HD, 4GB,

Adauga in cos

Verifica stoc magazin

Adauga in wishlist

Compara

REDUCERE

100 lei

1.699,99

IN STOC

Laptop LENOVO IdeaPad 330S-14IKB, Intel Core i3-8130U până la 3.4GHz, 14" Full HD,

Adauga in cos

Verifica stoc magazin

Adauga in wishlist

Compara

2018

VI. EVENTUALE OPTIMIZĂRI ȘI UPGRADE-URI

Ca optimizări pot fi incluse: realizarea unei interfețe grafice pentru a ușura utilizarea aplicației, posibilitatea de a salva într-un tabel sau într-o bază de date la un anumit interval de timp prețurile produselor pentru a realiza o statistică cu evoluția acestora, filtrarea notebook-urilor după anumite criterii alese de utilizator, afișarea recenziilor produselor și extinderea aplicației pentru toate categoriile de produse.

VII. BIBLIOGRAFIE

Site-uri web:

1. view-source:<https://altex.ro/laptopuri/cpl/>
2. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
3. <https://docs.python.org/3/library/urllib.request.html>
4. <https://docs.python.org/3/library/csv.html>
5. <https://ro.wikipedia.org/wiki/Python>
6. <https://www.crummy.com/software/BeautifulSoup/bs3/download/2.x/documentation.html>
7. https://en.wikipedia.org/wiki/Web_scraping
8. <https://www.lucidchart.com/>
9. <http://webdataextraction.blogspot.com/2016/01/website-scraping-and-data-extraction.html>

Cărți:

1. *Web Scraping with Python: Collecting Data from the Modern Web* (Ryan Mitchell, "O'Reilly Media, Inc.", 2015)