                        UNIVERSITETI I PRISHTINËS "HASAN PRISHTINA"

 FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE



**Raport**: “Zhvillimi i aplikacionit që mundëson URL and structure crawling dhe paraqitja e të njëjtave.”

**Lënda**: “Siguria në internet”

Grupi 29 - Detyra 7

                                                                                                  Studentët:

**Profesor**: Prof. Dr. Blerim Rexha                                          Natyra Vitija

**Asistent**: Msc. Arbnor Halili                                                   Tahir Temaj

                                                                                                  Teuta Ukshini

         Prishtinë, Shkurt 2021

**Përmbajtja**

Hyrje…………………………………………………………………………………………………….... 3

Përshkrimi i projektit………………………………………………………………………........................4

Kodi rreth projektit………………………………………………………………………………………..5

Testimi…………………………………………………………………………………………………….9

Konkluzioni……………………………………………………………………………………………….14

Referencat……………………………………………………………………………………................... 14

Hyrje

Në projektin e tretë,i cili është një project grupor në lënden Internet Security është dhënë detyra si vijon:

Aplikacioni duhet të përmbajë një meny për përzgjedhjen e opcioneve/funksionaliteteve të veta. Menyja duhet të jetë e punuar si GUI (graphical user interface) duke përdorur vetëm modulin tkinter të Python.

Aplikacionet duhet të shmangin përdorimin e moduleve të gatshme përvec në rastin kur një gjë e tillë është e pashmangshme.

Detyra 7

Zhvillimi i aplikacionit që mundëson URL and structure crawling dhe paraqitja e të njëjtave.

Projekti është zhvilluar me veglën Pycharm 2020.3.3 dhe gjendet ne github me repository Internet\_Security\_Gr.29 në linkun https://github.com/TeutaUkshini/Internet\_Security\_Gr.29



Repository është publik dhe përmban:



Përshkrim i projektit

Web scraping (Skrapimi i faqes) në internet, i quajtur shpesh web crawling ose spidering në internet, ose "kalimi programatik i një koleksioni faqesh në internet dhe nxjerrja e të dhënave", është një mjet i fuqishëm për të punuar me të dhënat në internet. Me një web scraper, ju mund të nxjerrni të dhëna për një grup produktesh, të merrni një korpus të madh teksti ose të dhëna sasiore për të luajtur, të merrni të dhëna nga një sajt(site) pa një API zyrtar, ose thjesht të kënaqni kuriozitetin tuaj personal. Në mënyrë ideale, një faqe në internet duhet të zvarritet e plotë (duke përfshirë çdo URL të lidhur në faqe). Sidoqoftë, faqet e internetit shumë të mëdha, ose faqet me shumë probleme arkitektonike, mund të mos jenë në gjendje të zvarriten plotësisht menjëherë. Mund të jetë e nevojshme të kufizoni zvarritjen në seksione të caktuara të faqes ose të kufizoni modelet specifike të URL-ve. Shkencëtarët e të dhënave duhet të dinë se si të mbledhin të dhëna nga faqet e internetit dhe t'i ruajnë ato të dhëna në formate të ndryshme për analiza të mëtejshme.

Kodi rreth projektit

Fajlli *index.py* përmban kodin rreth projektin në gjuhën programuese python në modulin tkinter.

Fillon me importimin e librarive dhe instalimi i dy librarive

> pip install pillow

> pip install beautifulsoup4

from tkinter import \*  
import tkinter as tk  
from tkinter import messagebox as ms  
from PIL import ImageTk, Image  
from bs4 import BeautifulSoup  
import urllib

Funkioni showHelp per komanden Help Guide nese dikush ka nevoje per te ditur me shume rreth projektit.

def showHelp(arg=None):  
 root = tk.Toplevel()  
 info = open('help.txt' , 'r').read()  
  
 # Krijimi i titullit  
 root.title('Help Guide')  
  
 # Krijimi i ikones(logo)  
 root.iconbitmap('img/logo.ico')  
  
 # Bllokimi i zgjerimit te madhesise se dritares(Window)  
 root.resizable(width=False, height=False)  
   
 ''' Perfundimi i dritares(Window)'''  
   
 # Shtimi i scrollbar ne dritare(window)  
 scrollbar = Scrollbar(root)  
 scrollbar.pack(side=RIGHT, fill=Y)  
  
 # Perdorimi i widget text per te shfaqur permbajtjen scraped  
 text = Text(root, yscrollcommand=scrollbar.set, wrap = WORD)  
 text.insert(INSERT, info)  
 text.pack()  
  
 # Scroll bar  
 scrollbar.config(command=text.yview)

Eshtë deklaruar funksioni Scape, ku bën kontrollimin e url-së nëse është i zbrazët të tregoj një mesazh gabimi, përndryshe fillon metoda Scraping me dhënjen e url-së.

def Scrape(arg=None):  
   
 # Kontrollimi i url-se nese eshte e zbrazet  
 if url\_entry.get() == '':  
 ms.showerror('Gabim', 'Shenoni nje URL valide !!!')  
   
 else:  
 try:  
 ''' Fillon metoda Scraping '''  
 # Dhenja e url-se  
 url = url\_entry.get()  
   
 # Leximi i tere permbajtjes  
 content = urllib.request.urlopen(url).read()  
   
 # Kalimi i permbajtjes tek funksioni  
 soup = BeautifulSoup(content, features="lxml")  
  
 # Ruajtja e html ne nje variabel  
 info = soup.prettify()  
 '''Perfundimi i metodes Scrape '''

Këtu behet hapja e një dritare të re,ku fillon me krijimin e titullit,vërjen e ikonës si dhe bllokimin i zgjerimit të madhësisë së asaj dritare.

'''Startimi i dritares(Window)'''  
# Krijimi i nje Dritareje(Window) te re  
root = tk.Toplevel()  
  
# Krijimi i titullit  
root.title('Falemnderit qe perdoret sherbimin tone !!!!')  
  
# Krijimi i ikones(logo)  
root.iconbitmap('img/logo.ico')  
  
# Bllokimi i zgjerimit te madhesise se dritares(Window)  
root.resizable(width=False, height=False)  
  
''' Perfundimi i dritares(Window)'''

Kodi per krijimin e menu-ve file me dropdown dhe help per help guide.

# Menu  
crawler\_menu = Menu(crawler)  
crawler.config(menu=crawler\_menu)  
  
# Menu items  
choose\_menu = Menu(crawler\_menu)  
help\_menu = Menu(crawler\_menu)  
  
# Menu objects  
crawler\_menu.add\_cascade(label="File" , menu=choose\_menu)  
choose\_menu.add\_command(label="URLs" , command=Scrape)  
choose\_menu.add\_command(label="Exit" , command=crawler.destroy)  
  
crawler\_menu.add\_cascade(label="Help" , menu=help\_menu)  
help\_menu.add\_command(label="Show" , command=showHelp)

Bëhet shtimi i scrollbar dhe përdorimi i widget text të përmbajtjes scraped

# Shtimi i scrollbar ne dritare(window)  
 scrollbar = Scrollbar(root)  
 scrollbar.pack(side=RIGHT, fill=Y)  
  
 # Perdorimi i widget text per te shfaqur permbajtjen scraped  
 text = Text(root, yscrollcommand=scrollbar.set, wrap = WORD)  
 text.insert(INSERT, info)  
 text.pack()  
  
 # Scroll bar  
 scrollbar.config(command=text.yview)  
  
except ValueError:  
 ms.showerror('Gabim', 'Shenoni nje URL valide !!!')

Këtu është bërë dizajnimi dhe bashkimi bind që të kthehet rezultati i pritur.

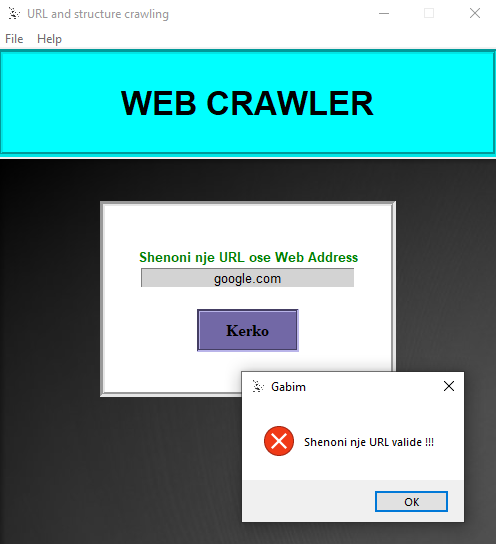
'''Startimi i dritares(Window)'''  
# Krijimi Widget  
crawler = tk.Tk()  
  
# Krijimi i madhesise se dritares(Window)  
crawler.geometry('500x500')  
  
# Bllokimi i madhesise se dritares(Window)  
crawler.resizable(width=False, height=False)  
  
# Krijimi i titullit  
crawler.title('URL and structure crawling ')  
  
# Krijimi i ikones(logo)  
crawler.iconbitmap('img/logo.ico')  
''' Perfundimi i dritares(Window) '''  
  
# Korniza(Frame) Kryesore  
top\_frame = Label(crawler, text='WEB CRAWLER',font = ('Cosmic', 25, 'bold'), bg='cyan', fg='black', relief='groove',padx=500, pady=30, bd='5')  
top\_frame.pack(side='top')  
  
''' Startimi i fotos ne prapavije(background)'''  
# Kalibrimi i madhesise se fotos  
canvas = Canvas(crawler, width=500, height=500)  
  
# Hapja e fotos  
image = ImageTk.PhotoImage(Image.open('img/bg6.jpg'))  
  
# Pozicionimi i fotos  
canvas.create\_image(0,0, anchor=NW, image=image)  
canvas.pack()  
'''Perfundimi i fotos ne prapavije(background)'''  
  
# Krijimi i Kornizes(Frame)  
frame = LabelFrame(crawler, padx=30, pady=40, bg='white', bd='5', relief='groove')  
frame.place(relx = 0.5, rely = 0.5, anchor = CENTER)  
  
# Etiketa(Label)  
url\_add = tk.Label(frame, text = 'Shenoni nje URL ose Web Address',font=('Arial',10, 'bold'),bg='white', fg='green').pack()  
  
# Hyrja ose Inputi  
url\_entry = tk.Entry(frame, font=('calibre',10,'normal'), justify = 'center', bg='lightgray', width='30')  
  
# Kthimi i nje vlere tek funksioni  
url\_entry.bind('<Return>', Scrape)  
  
# Fokuso tek inputi  
url\_entry.focus\_set()

Testimi

Pamja e parë e krijuar e dritarës që mundëson kërkimin e url-së së kërkuar.



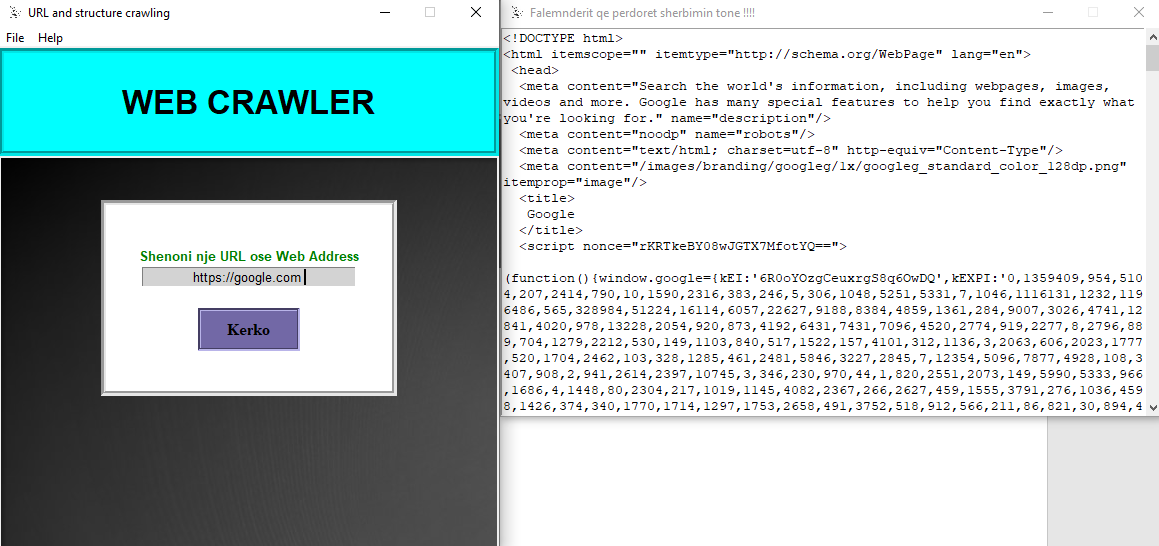
Kur bëhet pasimi i për kërkimin e url-së të shkruar gabim,shfaqet në ekran një mesazh gabimi me përshkrim që shënoni një url valide,si në pamjen në ekran.



Shënimi i url-se bëhet me domenin <https://google.com> në mënyrë që të shfaqet në ekran rezultati i kërkuar.



Pamja që shfaqet në ekran me anë të komandës Kerko,është rezultati i faqes google në html.

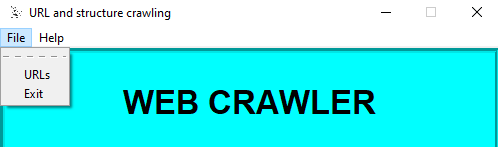


Pamja ne scroll bar

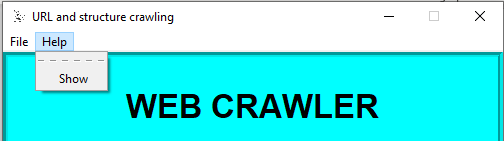


Menyte

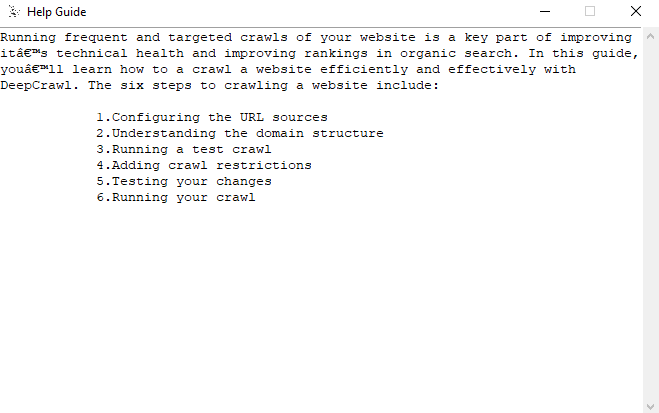
Krijimi i menyve file dhe help



Url ka funksionin e kerko,ndersa exit ben daljen nga program.



Help me shtypjen e Show shfaqet nje tab tjeter me text nga file help.txt si ne foton ne vijim:



Konkluzioni

* Kemi m\suar të kodojmë në gjuhën programuese python (duke përfshirë këtu modulin tkinter)
* Paraqitjen e GUI (graphical user interface) dhe punimi me të.
* Metodat Scraping dhe Crawling dhe përdorimi i tyre.
* Shfaqja e cdo url-je ne fajll Html.

Referencat

* <https://www.digitalocean.com/community/tutorials/how-to-crawl-a-web-page-with-scrapy-and-python-3>
* <https://www.freecodecamp.org/news/how-to-build-a-url-crawler-to-map-a-website-using-python-6a287be1da11/>
* <https://www.deepcrawl.com/knowledge/guides/how-crawl-website-with-deepcrawl/>
* <https://www.tutorialspoint.com/beautiful_soup/beautiful_soup_installation.htm>