

WYDZIAŁ
MATEMATYKI
I FIZYKI STOSOWANEJ
POLITECHNIKI RZESZOWSKIEJ

Projekt
Usługi sieciowe w biznesie

Scraper internetowy oraz obsługa bota spamerskiego

Przemysław Skubel
Inżynieria i analiza danych
rok III, sem. VI, L4, 160466

Rzeszów, 2022

Spis Treści

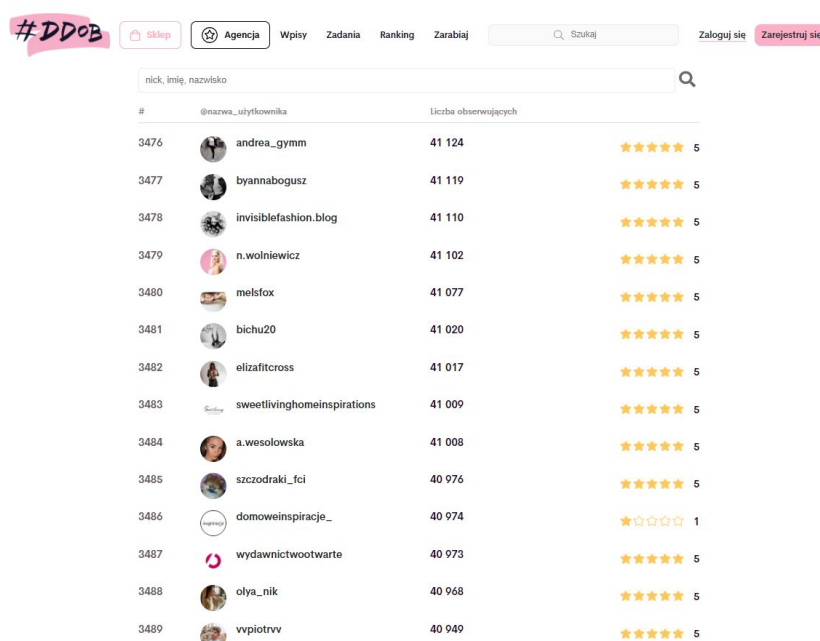
| | |
|--|----|
| 1. Opis projektu | 3 |
| 2. Zbieranie danych - Scraper | 3 |
| 3. Wysyłka wiadomości - Spamer | 8 |
| 4. Podsumowanie i wnioski | 11 |

1. Opis projektu

Projekt dotyczyć będzie usługi dla firm w celach marketingowych. Składa się on z dwóch części, pierwszej, która polegać będzie na zbieraniu danych dotyczących kont instagramowych influencerów ze strony internetowej za pomocą biblioteki Selenium w języku Python. W drugą część projektu wchodzi zaprojektowanie oraz implementacja bota, którego zadaniem będzie wysyłka wiadomości prywatnych o wybranej przez nas tematyce do zebranych wcześniej użytkowników za pomocą instagramowego API. Skrypt bota również zostanie napisany w języku Python. Środowiskiem, w którym będziemy pracować jest Visual Studio Code.

2. Zbieranie danych - Scraper

W internecie możemy znaleźć różne strony z bazami influencerów, jednak prawie wszystkie wymagają od nas wykupienia co najmniej miesięcznej subskrypcji. Istnieją również darmowe bazy, lecz są one mocno ograniczone na przykład nie znajdziemy w nich użytkowników o liczbie obserwujących jaka nas interesuje oraz na takich darmowych stronach nie mamy możliwości pobrania danych. W takich przypadkach wykorzystuje się scraping, w wolnym tłumaczeniu „zeskrobywanie informacji”. Strona, z której będziemy scrapować dane ma adres <https://ddob.com/ranking/instagram/> i wygląda następująco:



| # | @nazwa_uzytkownika | Liczba obserwujących | | |
|------|-----------------------------|----------------------|-------|---|
| 3476 | andrea_gymm | 41 124 | ★★★★★ | 5 |
| 3477 | byannabogusz | 41 119 | ★★★★★ | 5 |
| 3478 | invisiblefashion.blog | 41 110 | ★★★★★ | 5 |
| 3479 | n.wolniewicz | 41 102 | ★★★★★ | 5 |
| 3480 | melsfox | 41 077 | ★★★★★ | 5 |
| 3481 | bichu20 | 41 020 | ★★★★★ | 5 |
| 3482 | elizaftcross | 41 017 | ★★★★★ | 5 |
| 3483 | sweetlivinghomeinspirations | 41 009 | ★★★★★ | 5 |
| 3484 | a.wesolowska | 41 008 | ★★★★★ | 5 |
| 3485 | szczodraki_fci | 40 976 | ★★★★★ | 5 |
| 3486 | domoweinspiracje_ | 40 974 | ★☆☆☆☆ | 1 |
| 3487 | wydanictwootwarte | 40 973 | ★★★★★ | 5 |
| 3488 | olya_nik | 40 968 | ★★★★★ | 5 |
| 3489 | vpiotrrv | 40 949 | ★★★★★ | 5 |

Nasza strona internetowa jest nietrudnej budowy, wszystkie niezbędne informacje o kontach znajdują na jednej podstronie co bardzo ułatwia sprawę. Oto jak przedstawia się gotowy skrypt scrapera:

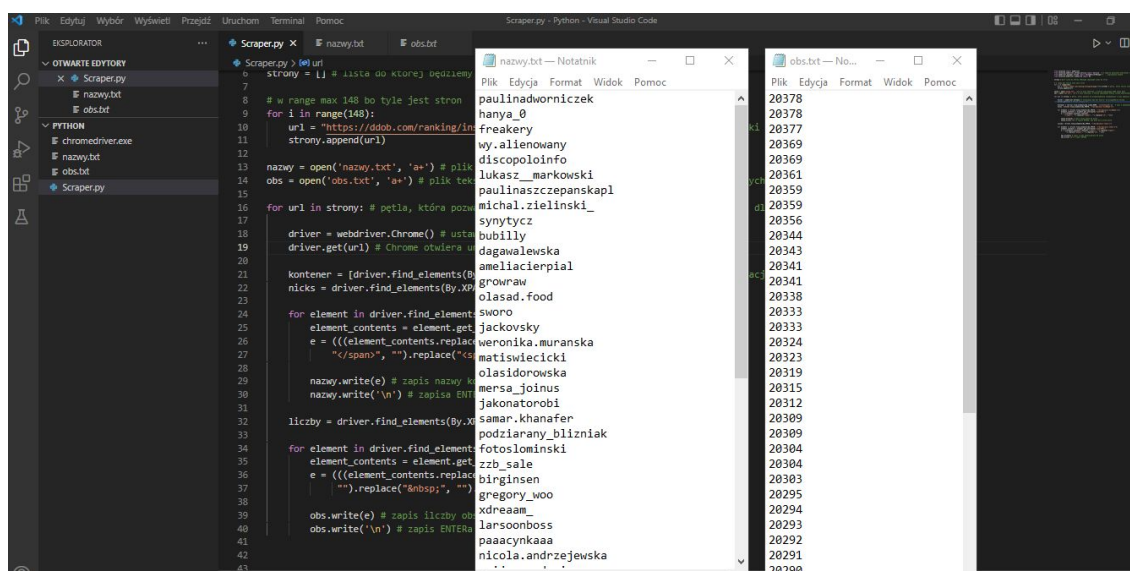
```
Scrapy X
1 from selenium import webdriver
2 from selenium.webdriver.chrome.service import Service # 4 importy dotyczące biblioteki selenium, które pozwolą nam na krótsze...
3 from webdriver_manager.chrome import ChromeDriverManager # ...zapisywanie komend
4 from selenium.webdriver.common.by import By
5
6 strony = [] # lista do której będziemy zapisywać linki do stron
7
8 # w range max 100 bo tyle jest stron
9 for i in range(140):
10     url = "https://ddob.com/ranking/instagram/page/" + str(i+250) # pętla, która tworzy linki do strony, z której zbieramy dane
11     strony.append(url)
12
13 nazwy = open('nazwy.txt', 'a') # plik tekstowy, w którym zapisywane będą nazwy kont
14 obs = open('obs.txt', 'a') # plik tekstowy, w którym zapisywane będzie liczba obserwujących konto danego influencera
15
16 for url in strony: # pętla, która pozwala na przeprowadzenie znajdujących w niej operacji dla każdej podstrony
17
18     driver = webdriver.Chrome() # ustawienie aby bot działał na przeglądarce Chrome
19     driver.get(url) # Chrome otwiera url
20
21     kontener = driver.find_elements(By.CLASS_NAME, 'listContainer') # opis w dokumentacji
22     nicks = driver.find_elements(By.XPATH, "//div[@class='nickname']")
23
24     for element in driver.find_elements(By.XPATH, "//div[@class='nickname']"):
25         element_contents = element.get_attribute('innerHTML')
26         e = ((element_contents.replace(
27             "<span>", "").replace("<span>", "").replace("<div>", "")))
28         nazwy.write(e) # zapis nazwy konta do pliku
29         nazwy.write("\n") # zapis ENTERA, aby dane były przeliny
30
31     liczby = driver.find_elements(By.XPATH, "//div[@class='stats']")
32
33     for element in driver.find_elements(By.XPATH, "//div[@class='stats']"):
34         element_contents = element.get_attribute('innerHTML')
35         e = ((element_contents.replace("<span>", "").replace("<div>", "").replace("<div>", "")))
36         obs.write(e) # zapis liczby obserwujących do pliku
37         obs.write("\n") # zapis ENTERA
38
39
40
41
```

Za pomocą inspekcji dostępnej w każdej przeglądarce, mamy dostęp do kodu HTML strony, na której się aktualnie znajdujemy. Uruchamiamy ją za pomocą skrótu klawiszowego ctrl + shift + i, lub po prostu klikając prawym przyciskiem myszki w dowolnym miejscu na stronie i wybierając opcję 'Zbadaj'. Dzięki temu możemy znaleźć interesującą nas treść ze strony w kodzie źródłowym HTML. Na początek wyszukujemy „kontenera”, w którym znajdują się nazwy użytkownika oraz liczba obserwatorów. Wygląda to następująco:

The screenshot shows a web browser window with the URL <https://ddob.com/ranking/instagram/page/140>. The page displays a ranking of Instagram accounts. A right-click context menu is open over the table, showing options like 'Ta strona korzysta z ciasteczek'. The browser's developer tools are open on the right, showing the HTML structure of the page. The table contains the following data:

| # | @nazwa_uzytkownika | Liczba obserwujących | | |
|------|-----------------------|----------------------|-------|---|
| 3476 | andrea_gymm | 41 124 | ★★★★★ | 5 |
| 3477 | byannabogusz | 41 119 | ★★★★★ | 5 |
| 3478 | invisiblefashion.blog | 41 110 | ★★★★★ | 5 |
| 3479 | n.wolniewicz | 41 102 | ★★★★★ | 5 |
| 3480 | meisfox | 41 077 | ★★★★★ | 5 |
| 3481 | a.wesolowska | 41 038 | ★★★★★ | 5 |
| | | 41 020 | ★★★★★ | 5 |
| | | 41 017 | ★★★★★ | 5 |

Używając ctrl+f w inspekcji możemy wyszukiwać frazy w kodzie HTML. Wyszukiwanie klas i tym podobnie następuje poprzez wpisanie XPatha lub Selektora. Ja będę korzystać z tego pierwszego. W 21 linijce kodu do zmiennej **kontener** przypisujemy tą część HTMLa, która jest klasą o nazwie **listContainer**. Podobnie dla nazw użytkowników, linijka nr 22, do zmiennej **nicks** przypisujemy za pomocą klasę **nickName**. Następnie tworzymy pętlę, która będzie przeszukiwać klasę **nickName** oraz wyciągnie i zapisze do pliku nazwę użytkownika. Analogicznie pętla zaczynająca się w 34 linijce będzie wyszukiwać oraz zapisywać liczbę obserwowanych. Oto jak przedstawiają się zebrane dane z dwóch stron:



The screenshot shows a Visual Studio Code editor with three open files. The leftmost file is 'Scrapy.py', which contains a Python script using Selenium to scrape data from a website. The script iterates through a range of 148 URLs, finds elements with class 'nickName' and 'listContainer', and writes the extracted names and follower counts to two text files: 'nazwy.txt' and 'obs.txt'. The rightmost file is 'nazwy.txt', which contains a list of usernames. The middle file is 'obs.txt', which contains a list of follower counts.

```
Scrapy.py
1 # -*- coding: utf-8 -*-
2 import sys
3 import os
4 import time
5 from selenium import webdriver
6 from selenium.webdriver.common.by import By
7 from selenium.webdriver.support.ui import WebDriverWait
8 from selenium.webdriver.support import expected_conditions as EC
9
10 # w range max 148 bo tyle jest stron
11 for i in range(148):
12     url = "https://dodob.com/ranking/instrony.append(url)
13
14     nazwy = open('nazwy.txt', 'a') # plik
15     obs = open('obs.txt', 'a') # plik
16
17     for url in strony: # pętla, która poz
18         driver = webdriver.Chrome() # usta
19         driver.get(url) # Chrome otwiera u
20
21         kontener = [driver.find_elements(B
22         nicks = driver.find_elements(By.XP
23
24         for element in driver.find_element
25             element_contents = element.get
26             e = (((element_contents.replac
27                 "</span>", "").replace("<
28             nazwy.write(e) # zapis nazwy k
29             nazwy.write("\n") # zapis E
30
31         liczby = driver.find_elements(By.X
32
33         for element in driver.find_element
34             element_contents = element.get
35             e = (((element_contents.replac
36                 "").replace("&nbsp;", ""
37             obs.write(e) # zapis liczby ob
38             obs.write("\n") # zapis E
39
40         time.sleep(10)
41
42         driver.quit()
43
44 nazwy.txt
45 paulinadworniczek
46 hany_0
47 freakery
48 wy.allenowany
49 discopoloinfo
50 lukasz_markowski
51 paulinaszczepanskapl
52 michal.zielinski_
53 synyttycz
54 bubilly
55 dagawalewska
56 ameliaciempial
57 growraw
58 olasad.food
59 sworo
60 jackovsky
61 weronika.muranska
62 matiswiecicki
63 olasidorowska
64 mersa_joinus
65 jakonatorobi
66 samar.khanafar
67 podziarany_blizniak
68 fotoslominski
69 zzb_sale
70 birginsen
71 gregory_woo
72 xdream
73 larsoonboss
74 paaacynkaaa
75 nicola.andrzejewska
76
77 obs.txt
78 20378
79 20378
80 20377
81 20369
82 20369
83 20361
84 20359
85 20359
86 20356
87 20344
88 20343
89 20341
90 20341
91 20338
92 20333
93 20333
94 20324
95 20323
96 20319
97 20315
98 20312
99 20309
100 20309
101 20304
102 20304
103 20303
104 20295
105 20294
106 20293
107 20292
108 20291
109 20290
```

Patrząc w przyszłość, aby nasz bot Spamer był bardziej wiarygodny przy pisaniu wiadomości, napisałem małego bota, który zbierze informacje o imionach użytkowników. Niestety, nie każdy użytkownik posiada dane o swoim imieniu zatem na potrzeby projektu użyjemy tylko tych użytkowników, których imię udało się zdobyć (baza z imionami liczy 1304 użytkowników). Aby zbierać imiona za pomocą krótkiej pętli utworzyłem plik z linkami do profili. Niestety nie wiem od czego to zależy, ale czasami instagram poprosi nas o zalogowanie aby wyświetlić profil a czasami nie. Poniżej kod do z tworzeniem linków oraz kod do zbierania imion:

```

1 nazwy = open("nazwy.txt", 'r')
2 open("linki.txt", 'r')
3
4 for nazwa in nazwy:
5     link = 'https://www.instagram.com/'+nazwa
6     linki.txt.write(link)
7     linki.txt.write('\n')
8
9

```

```

1 from selenium import webdriver
2 from selenium.webdriver.chrome.service import Service
3 from webdriver_manager.chrome import ChromeDriverManager
4 from selenium.webdriver.common.by import By
5 from selenium.webdriver.support.ui import WebDriverWait
6 import time
7 import pandas as pd
8 import csv
9
10 links = open('linki.txt', 'r')
11 plik = links.read().split('\n')
12 session = open('session.txt', 'w')
13
14 for url in plik:
15     driver = webdriver.Chrome()
16     driver.get(url)
17     WebDriverWait(driver, 5)
18     acceptacja = driver.find_element(By.XPATH, '/html/body/div[4]/div/div/button[1]')
19     acceptacja.click()
20     time.sleep(4)
21     try:
22         data = driver.find_element(By.XPATH, "//div[@class='QOPtr']/div")
23         data = data.get_attribute('innerHTML')
24         print(data)
25         session.write(data)
26         session.write('\n')
27     except:
28         # print('Nie udało się')
29         # driver.close()
30         session.write('-----')
31         session.write('\n')

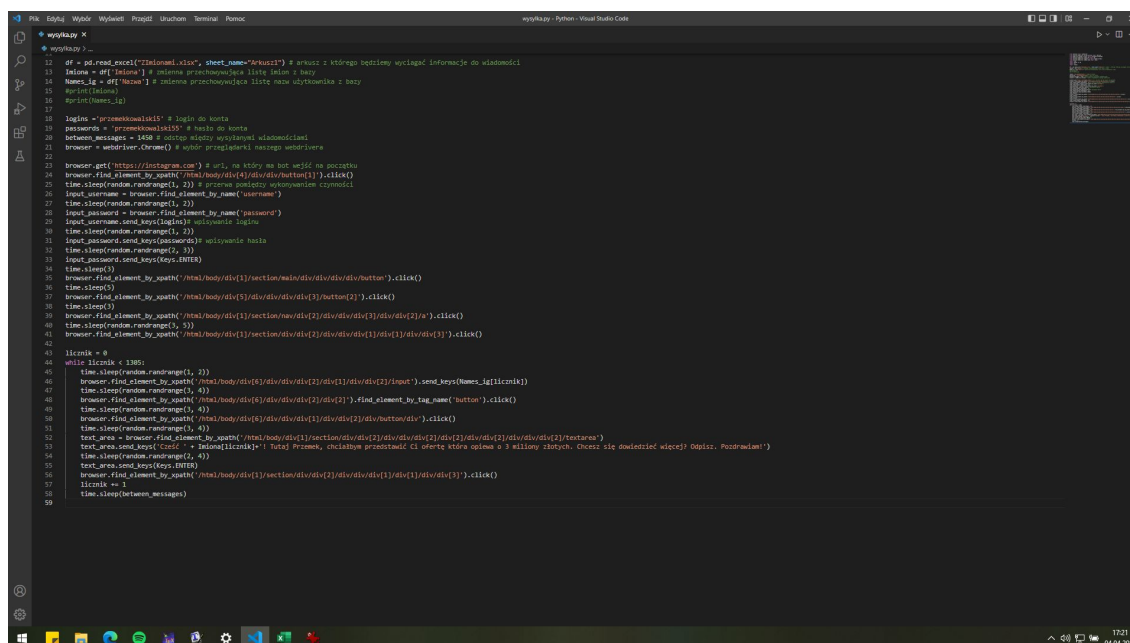
```

Wszystkie dane zostały zebrane do plików txt, zatem za pomocą importu wrzuciłem je do jednego arkusza Excel. Oto jak przedstawia się baza danych:

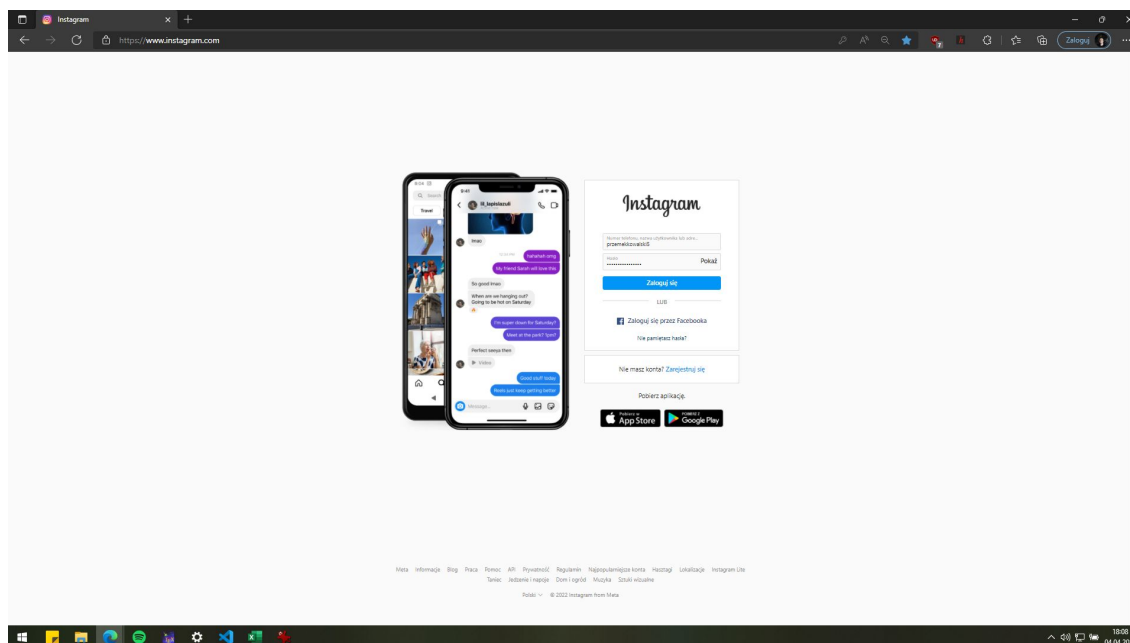
| link | Nazwa | Identyfikator | Imię |
|--|------------------------------|---------------|------------|
| https://www.instagram.com/https.skbl | https.skbl | 580 | Przemek |
| https://www.instagram.com/nworo | nworo | 50287 | Aleksandra |
| https://www.instagram.com/makadentystka | makadentystka | 50257 | Karolina |
| https://www.instagram.com/erwin_kubik | erwin_kubik | 50243 | Erwin |
| https://www.instagram.com/spzerek | spzerek | 50243 | Grzegorz |
| https://www.instagram.com/patryk_puczykowski | patryk_puczykowski | 50236 | Patryk |
| https://www.instagram.com/kiszkowscy_swiat_moniki | kiszkowscy_swiat_moniki | 50219 | Monika |
| https://www.instagram.com/olasad.food | olasad.food | 50214 | Aleksandra |
| https://www.instagram.com/kingspaulinajaz | kingspaulinajaz | 50205 | Kinga |
| https://www.instagram.com/cosmetic_variations | cosmetic_variations | 50199 | Weronika |
| https://www.instagram.com/synitycz | synitycz | 50196 | Monika |
| https://www.instagram.com/annadziubek_bydziubeka | annadziubek_bydziubeka | 50181 | Anna |
| https://www.instagram.com/remo_motywuje | remo_motywuje | 50172 | Remigiusz |
| https://www.instagram.com/marolischin | marolischin | 50171 | Mario |
| https://www.instagram.com/radek_majeczki | radek_majeczki | 50166 | Radek |
| https://www.instagram.com/melodylaniella | melodylaniella | 50159 | Anita |
| https://www.instagram.com/qqiaa | qqiaa | 50157 | Anna |
| https://www.instagram.com/tomek.dvorniak | tomek.dvorniak | 50152 | Tomek |
| https://www.instagram.com/matiswiecicki | matiswiecicki | 50144 | Mateusz |
| https://www.instagram.com/slittek | slittek | 50117 | Sławek |
| https://www.instagram.com/paulinawyska | paulinawyska | 50105 | Paulina |
| https://www.instagram.com/dobrukarolina | dobrukarolina | 50104 | Karolina |
| https://www.instagram.com/natalina1 | natalina1 | 50087 | Natalia |
| https://www.instagram.com/krzysztofamraczy | krzysztofamraczy | 50075 | Krzysztof |
| https://www.instagram.com/dominika.mindsee | dominika.mindsee | 50063 | Dominika |
| https://www.instagram.com/leahak | leahak | 50058 | Anna |
| https://www.instagram.com/migottka | migottka | 50045 | Sara |
| https://www.instagram.com/leilaartist_kreatorwizerunku | leilaartist_kreatorwizerunku | 50042 | Leila |
| https://www.instagram.com/naroya_czartoryska | naroya_czartoryska | 50040 | Naroya |
| https://www.instagram.com/daria.miko | daria.miko | 50030 | Daria |
| https://www.instagram.com/iga_przybylska | iga_przybylska | 50017 | Iga |
| https://www.instagram.com/lukaszguz | lukaszguz | 50013 | Lukasz |
| https://www.instagram.com/k_dawidowiczka | k_dawidowiczka | 50003 | Karolina |
| https://www.instagram.com/ewelinaagralak | ewelinaagralak | 19993 | Ewelina |
| https://www.instagram.com/zmyslowo | zmyslowo | 19975 | Magdalena |
| https://www.instagram.com/maciek_97 | maciek_97 | 19968 | Maciek |

3. Wysyłka wiadomości - Spamer

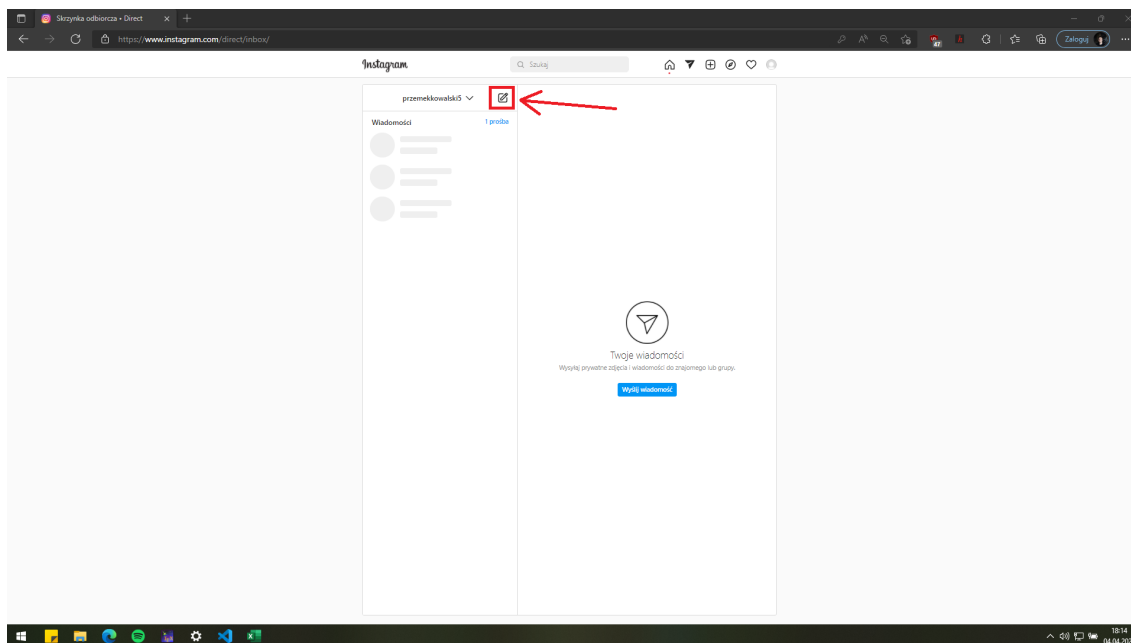
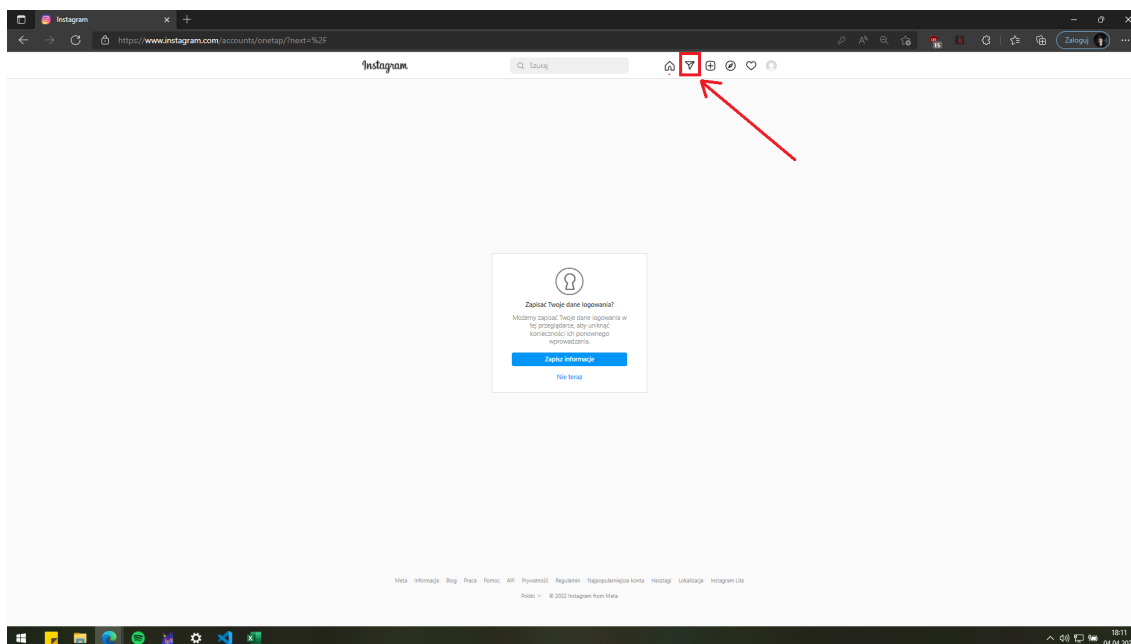
Posiadając wszystkie niezbędne informacje, możemy zaimplementować bota Spamera. Poniżej skrypt programu.



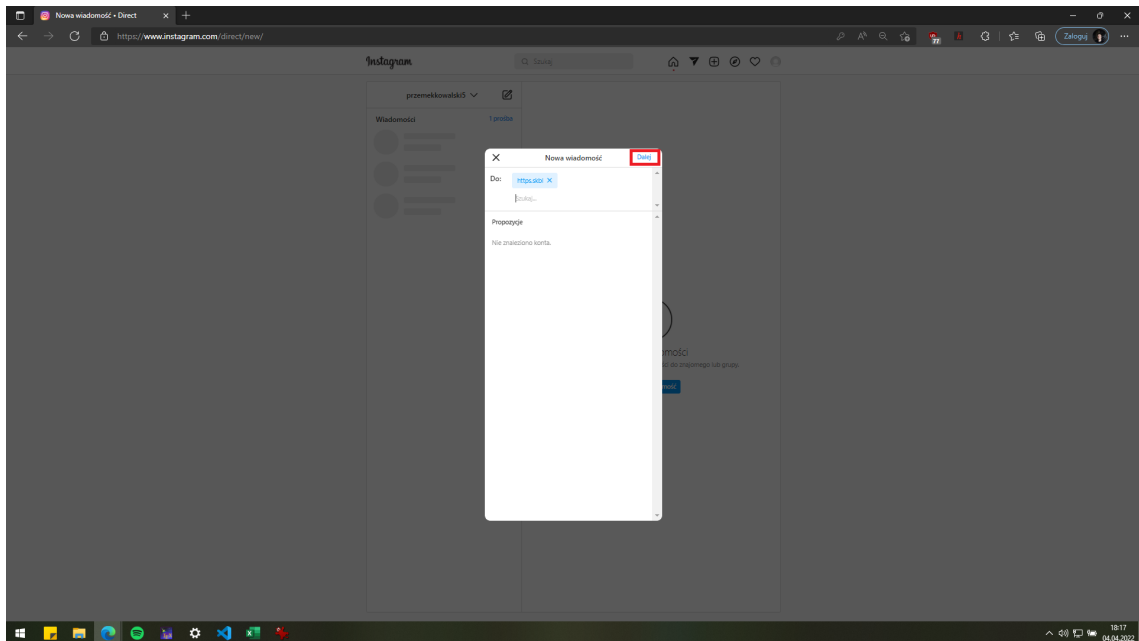
Na potrzeby projektu stworzyłem nowe konto, z którego będę prowadzić wysyłkę. Spamer otwiera przeglądarkę Chrome i przełącza się na stronę instagrama. Za pomocą XPath lokalizowane na stronie są miejsca do wpisania loginu i hasła. (dane do logowania zawarte są w skrypcie)



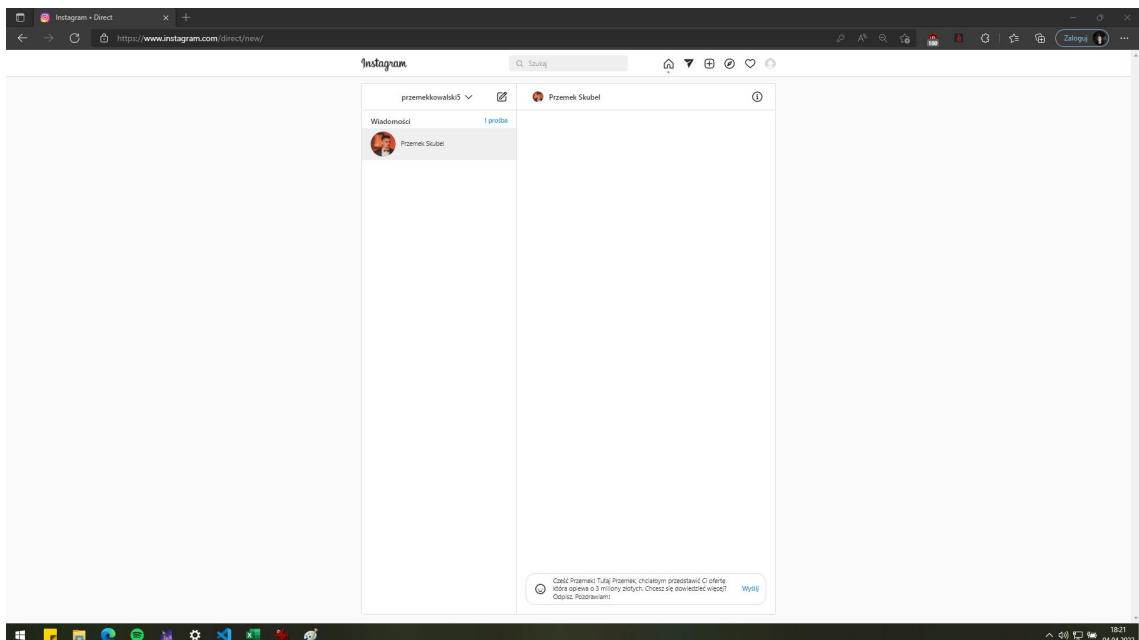
Po wpisaniu ich, zostajemy przekierowani na stronę główną. Następnie bot wchodzi w zakładkę 'wiadomości', a tam klika w ikonę utworzenia nowej konwersacji.



Następnie po zlokalizowaniu kontenera do wpisania nazwy użytkownika, wypełniamy go nazwą z listy o pozycji 'licznik'. Klikając 'Dalej', zostajemy przekierowani na konwersację z danym użytkownikiem.



Analogicznie wyszukujemy kontenera na wpisanie wiadomości. Bot wpisuje wiadomość, którą wcześniej napisaliśmy w skrypcie, a następnie wysyła ją klikając 'Wyślij'.



Cały proces wysyłania znajduje się w pętli.

4. Podsumowanie i wnioski

Jak możemy zauważyć, w kodzie ustawiona jest duża odległość między wysyłanymi wiadomościami. Jest to spowodowane tym, że instagram pozwala na utworzenie 60 konwersacji dziennie. Jeśli ustawimy 1440 sekund przerwy między wysłaniem wiadomości, pozwoli to na działanie bota cały czas w tle przez wiele dni, dopóki wiadomość nie zostanie wysłana na wszystkie konta. Cały proces wysyłania znajduje się w pętli. Taki bot pozwala na sprawną, jednakże w wielu przypadkach niechcianą reklamę na przykład danego produktu lub usługi. Przedsiębiorstwa często korzystają z takich rozwiązań, ponieważ są one tańsze od wykupienia reklamy na danym portalu. Data Scraping jest często używany podczas tworzenia dużych baz i hurtowni danych.