

**COMPUTER ENGINEERING DEPARTMENT**

**Web Data Mining**

Final Homework: LinkedIn Scraper

Hasibullah Mahmood - 160709073

# **Introduction**

In this Homework, We developed a scraper bot that establishes a session with LinkedIn Website. Then, it searches for a query on Google. After that, it fetches all LinkedIn profiles URLs from the Google page. Then, it scrapes each profile on LinkedIn and fetches his name, address, and contact information. Finally, the script saves the gathered data on the CSV file.

Used Python Packages:

* + - selenium
    - selenium.webdriver
    - parsel
    - pandas
    - time

Teams Tasks:

|  |  |
| --- | --- |
| Ahmed H. Ibrahim | Hasibullah Mahmood |
| * Establishing autonomous Session with LinkedIn | * Querying Google and Scarping profile URLs from it. |
| * Scarping LinkedIn Profiles data. | * Manages data storage process and Saving Data to CSV |
| * + Creating Parameters file with many Variables. | * + Creating Person Class. |

# **Names of the files with a short explanation**

***seleniumScraper.py*** – the main script that establishes a LinkedIn session, queries Google, scraps Profiles, saves data to CSV file.

***Person.py*** – A class that is imported by seleniumScarper.py to create an object for each profile.

***Parameters.py*** – A python file that has many variables that are imported by seleniumScraper.py

# **Methods (with their signature) in seleniumScraper.py:**

1. establishLinkedInSession(*in\_email*,*in\_password*) **--- it takes LinkedIn email and password as arguments. Then, it automates the login process with LinkedIn.**

def establishLinkedInSession(*in\_email*,*in\_password*):

*#### start 1. Making a new login session with linkedin*

*# Login to LinkedIn*

    driver.get(parameters.linkedIn\_login\_page)

*# Find Email Field and send email info.*

    email = driver.find\_element\_by\_id('username')

    email.send\_keys(in\_email)

    time.sleep(3)

*# Find password Field and send password info.*

    password = driver.find\_element\_by\_id('password')

    password.send\_keys(in\_password)

    time.sleep(3)

*# Find Submit Button and Click it.*

    logInBtn = driver.find\_element\_by\_xpath('//\*[@type="submit"]')

    logInBtn.click()

    time.sleep(5)

1. queryGoogle(*query*,*numOfPages*) **--- it takes the query string and numOfResults as arguments. numOfResults = number of profiles, you want to scrap. Then, it fetches all URLs from google to an array.**

def queryGoogle(*query*,*numOfResults*):

    driver.get(parameters.googleUrl)

    time.sleep(5)

*# Find Search input field and send search info to it.*

    searchingField=driver.find\_element\_by\_name('q')

    searchingField.send\_keys(query)

    time.sleep(5)

    searchingField.send\_keys(Keys.RETURN)

    time.sleep(5)

    driver.get(driver.current\_url + numOfResults)

*# Get Profiles Urls from Google querying result*

    for rapper in driver.find\_elements\_by\_class\_name('r'):

        for a in rapper.find\_elements\_by\_xpath('.//a'):

            search\_urls\_holder.append(a.get\_attribute('href'))

    print(search\_urls\_holder)

    time.sleep(2)

1. getProfileInfo(*profile\_url*) **--- This function will be called each time to scrape each profile data. It will scrape the name, address, and Contact Information. Contact Information may include ‘Twitter Url’, ‘Email Address’, ‘Blog Url’, and many other links. It will create an object for this profile and saves profile data in it. Also, it appends the cleaned data to many arrays that will be group in a CSV data frame later.**

def getProfileInfo(*profile\_url*):

*# Session with linkedin profile*

    html\_page=driver.get(profile\_url)

    time.sleep(3)

*# Get page Source*

    selector = Selector(*text*=driver.page\_source)

*# Get person name*

    name = selector.xpath(".//ul[contains(@class,'pv-top-card--list')]/li[contains(@class,'t-24')]/text()").get()

*# Get person address*

    address = selector.xpath(".//ul[contains(@class,'pv-top-card--list')]/li[contains(@class,'t-16')]/text()").get()

    time.sleep(4)

*# Find contact info Button and Click it.*

    contactInfoBtn = driver.find\_element\_by\_xpath('//\*[@data-control-name="contact\_see\_more"]')

    contactInfoBtn.click()

    time.sleep(4)

*# Update the selector to fetch contact information*

    selector = Selector(*text*=driver.page\_source)

    contact\_links = selector.css('.pv-contact-info\_\_contact-link \*::attr(href)').getall()

*# Outputting*

    p\_name = name.strip()

    p\_address = address.strip()

    p1 = Person(profile\_url, p\_name, p\_address, contact\_links)

*# Adding Person info to csv dataframe arrays*

    profile\_urls\_holder.append(profile\_url)

    names\_holder.append(p1.name)

    addresses\_holder.append(p1.address)

    contact\_info\_holder.append(p1.listToString())

1. saveProfilesData(*csvFile*,*urls*,*names*,*addresses*,*contact\_infos*): **--- it takes csvFile path, URLs array, names array, addresses array, contact\_infos array as arguments. Then, it will create a data frame that groups all of the arrays. Then, it will write the data frame to the CSV file.**

def saveProfilesData(*csvFile*,*urls*,*names*,*addresses*,*contact\_infos*):

*# Writing dataframe that combines arrays to csv file*

*# dictionary of lists*

    dict = {'Url': urls, 'Name': names, 'Address': addresses, 'Contact Info': contact\_infos}

    df = pd.DataFrame(dict)

*# saving the dataframe*

    df.to\_csv(csvFile, *index*=False, *encoding*='utf-8-sig')

# **Person Class in Person.py:**

class Person:

    def \_\_init\_\_(*self*, *url*, *name*, *address*, *contactInfo*):

        self.url = url

        self.name = name

        self.address = address

        self.contactInfo = contactInfo

    def listToString(*self*):

*# initialize an empty contact Info links String holder.*

        links = ""

*# Adding Links to the String.*

        for link in self.contactInfo:

            links = links + link + " | "

*# return string*

        return links

1. **Variables in parameters.py:**

chromeWebDriverPath = 'D:/Programs/chromedriver'

linkedIn\_login\_page = 'https://www.linkedin.com/login?fromSignIn=true&trk=guest\_homepage-basic\_nav-header-signin'

googleUrl = 'https://www.google.com'

*# login details*

in\_email = 'example@gmail.com'

in\_password = 'linkedInPassword'

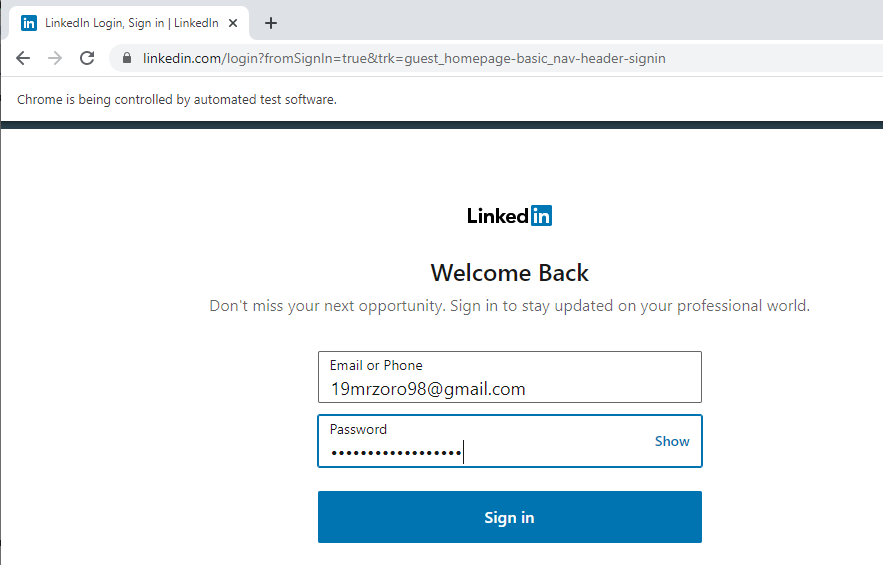
*# Search details*

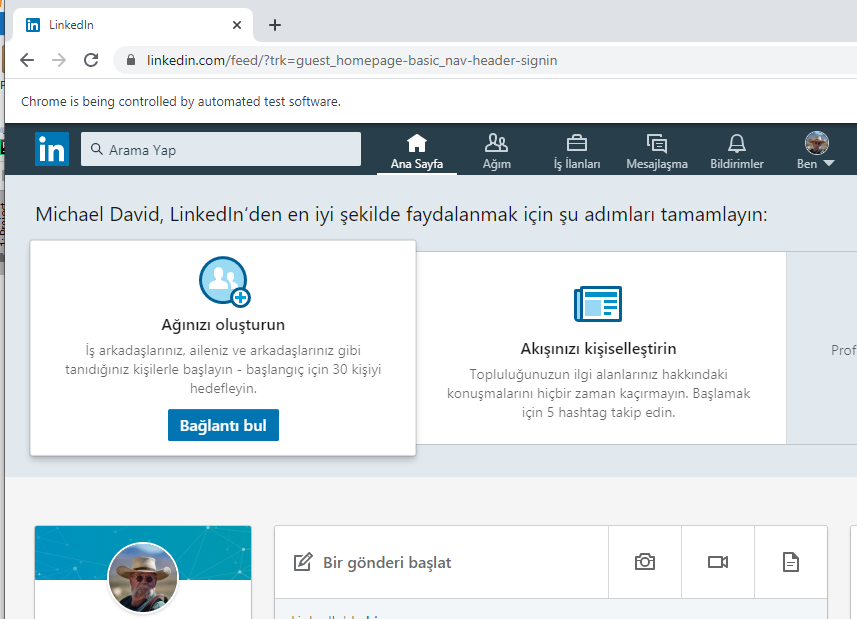
google\_search\_query = 'site:linkedin.com/in/ AND "student" AND "Izmir"'

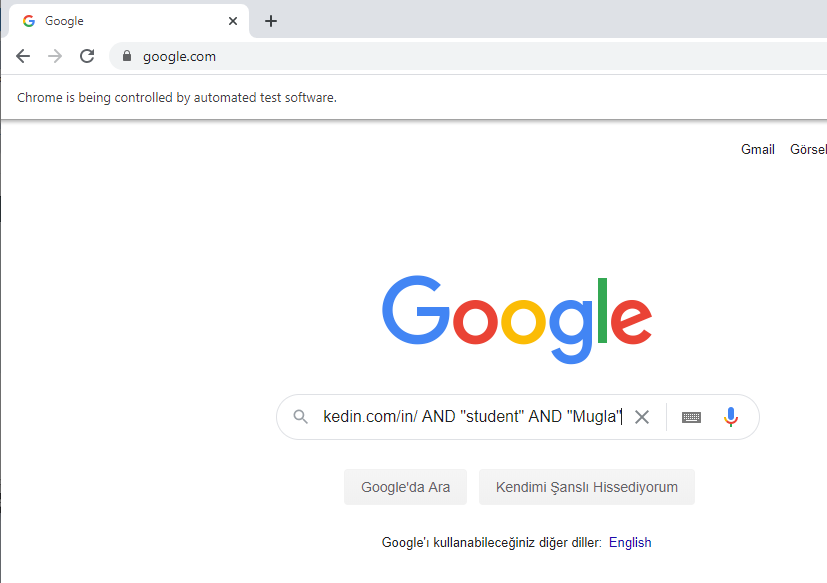
*# CSV file name*

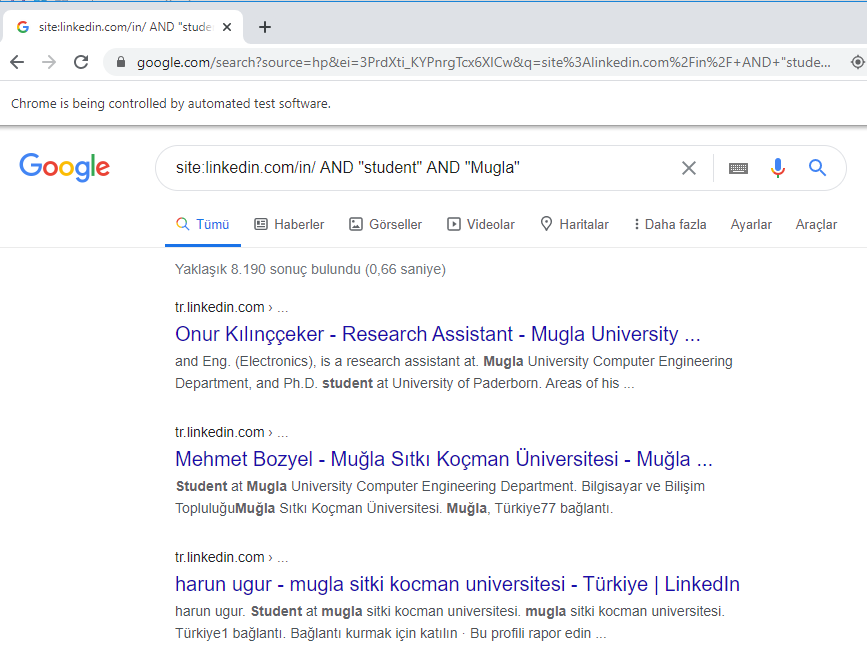
csv\_file\_name='LinkedIn\_Profiles\_Info.csv'

1. **sample output/screenshots:**
2. Automatic Login

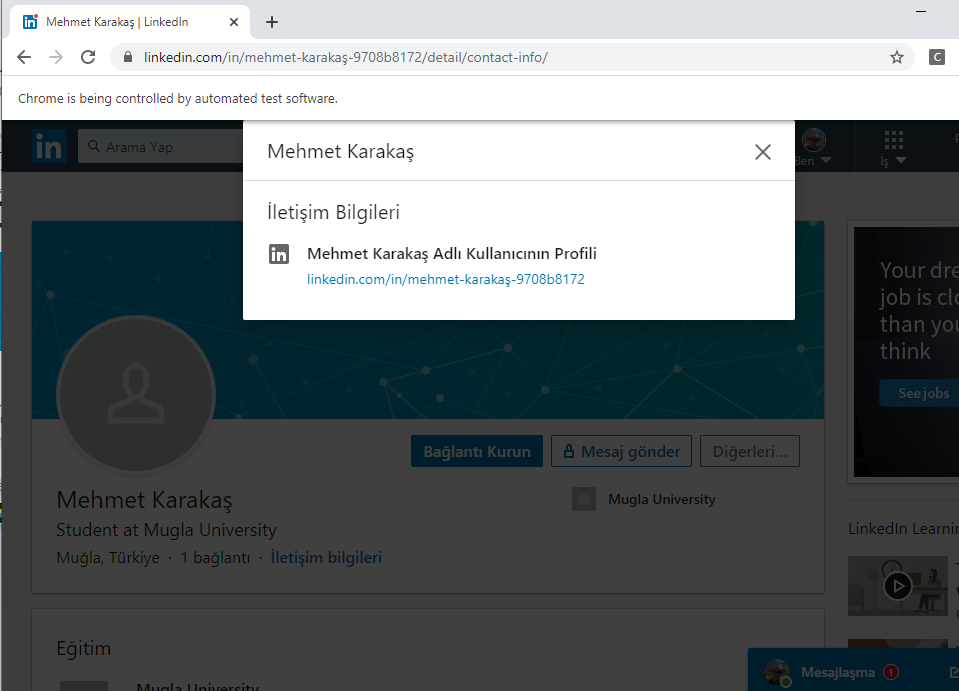


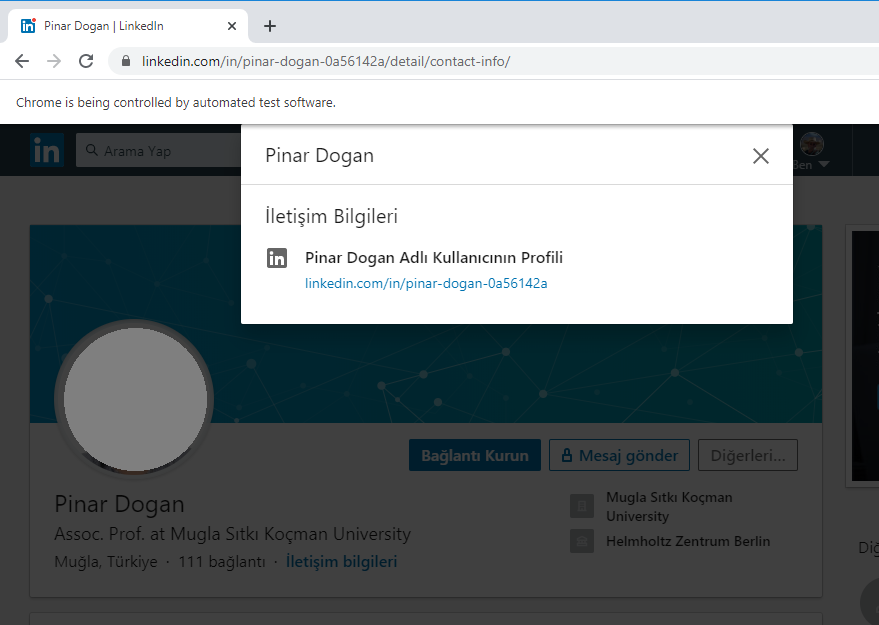


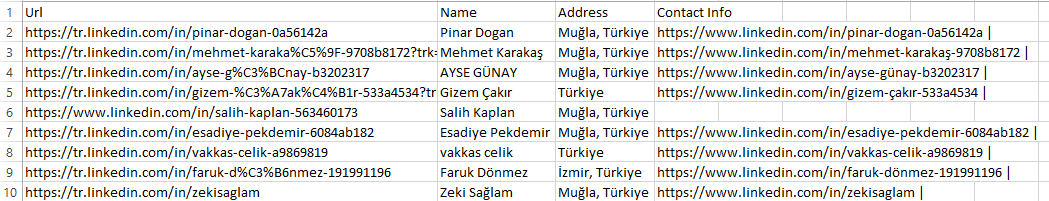
1. **Automatic Querying Google**

****

1. **Scraping Data from gathered accounts URLs**





1. **Storing Gathered Data in CSV file**