# Project Description

Pada project ini saya melakukan proses scraping pada website Tokopedia. Karena website Tokopedia adalah dynamic dan menggunakan JavaScript, tidak cukup menggunakan library Requests dan BeautifulSoup saja untuk melakukan scraping. Sehingga pada project ini saya menggunakan library Selenium untuk melakukan scraping. Scraping menggunakan library Selenium membutuhkan suatu driver yang sesuai dengan browser yang akan digunakan untuk scraping.

# Script

# Import libraries

from selenium import webdriver

from selenium.webdriver.common.by import By

import pandas as pd

import time

from selenium.common.exceptions import NoSuchElementException

# Create an empty list to store the data

data = []

# Configure selenium

options = webdriver.ChromeOptions()

options.add\_argument("--headless=new")

driver = webdriver.Chrome(options=options)

# Loop through pages 1 to 10

for page in range(1, 4):

    # print status

    print("Please wait, scraping page {}".format(page))

    # Construct the page URL

    url = f"https://www.tokopedia.com/search?navsource=&ob=5&page={page}&q=laptop&srp\_component\_id=04.06.00.00&srp\_page\_id=&srp\_page\_title=&st="

    driver.get(url)

    time.sleep(10)

    # Scraping process

    products = driver.find\_elements(By.XPATH, "//div[@class='prd\_link-product-name css-3um8ox']")

    for product in products:

        product\_name = product.text

        parent = product.find\_element(By.XPATH, "..")

        try:

            price\_element = parent.find\_element(By.XPATH, ".//div[@class='prd\_link-product-price css-1ksb19c']")

            price = price\_element.text

        except NoSuchElementException:

            price = None

        try:

            location\_element = parent.find\_element(By.XPATH, ".//span[@class='prd\_link-shop-loc css-1kdc32b flip']")

            location = location\_element.text

        except NoSuchElementException:

            location = None

        try:

            rating\_element = parent.find\_element(By.XPATH, ".//span[@class='prd\_rating-average-text css-t70v7i']")

            rating = rating\_element.text

        except NoSuchElementException:

            rating = None

        try:

            sales\_element = parent.find\_element(By.XPATH, ".//span[@class='prd\_label-integrity css-1duhs3e']")

            sales = sales\_element.text

        except NoSuchElementException:

            sales = None

        data.append({

            'Product': product\_name,

            'Price': price,

            'Rating': rating,

            'Location': location,

            'Sales': sales,

        })

driver.quit()

print("Scraping process is done.")

# Create a DataFrame from the collected data

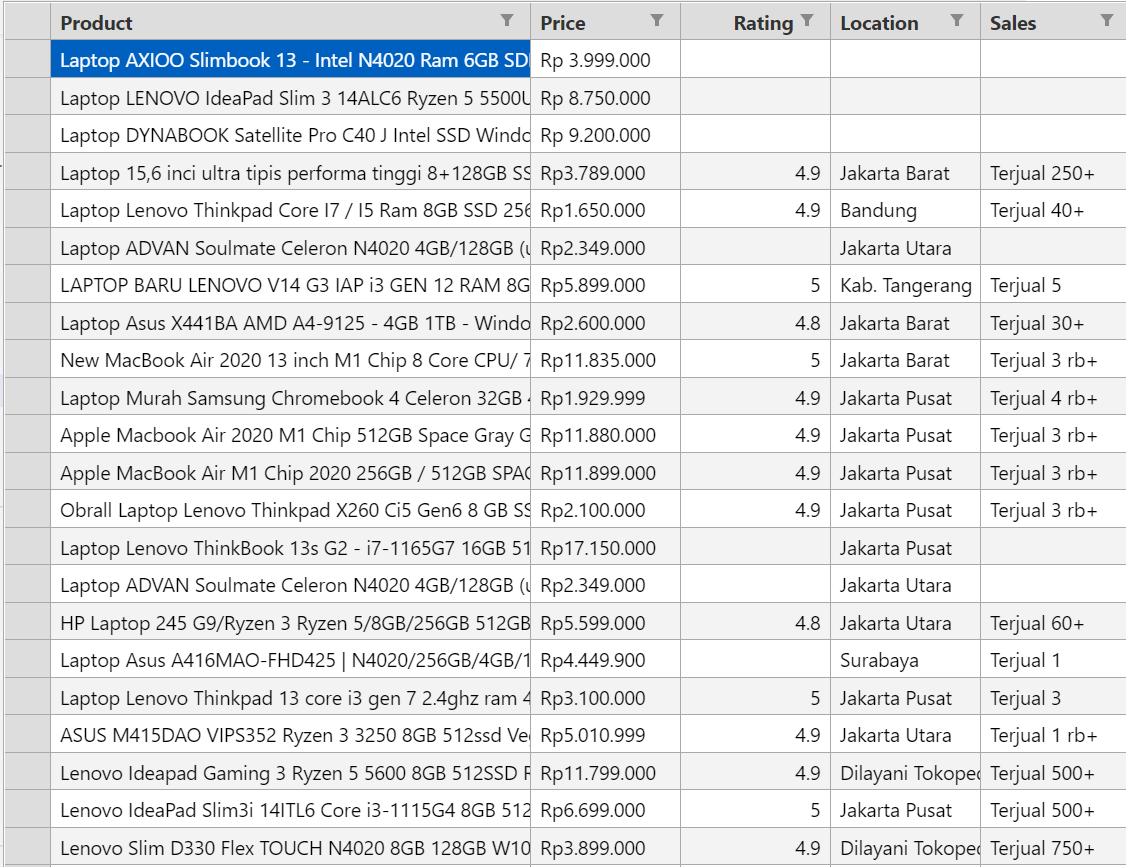
df = pd.DataFrame(data)

# Create a CSV file

filepath = r'D:\Personal Projects\web\_scraping\scraped\_tokopedia\_data.csv'

df.to\_csv(filepath, index=False)

# Output



# Github Repo

<https://github.com/adrn-mm/web_scraping>

# References

* [Web scraping javascript using Python](https://towardsdatascience.com/data-science-skills-web-scraping-javascript-using-python-97a29738353f)
* [Web scraping with Python in Indonesian E-Commerce](https://medium.com/@yohan.ardiansyah90/web-scraping-with-python-in-indonesian-e-commerce-tokopedia-part-1-getting-the-data-a338ebd56306)