SHERAZ BIN TAHIR

Mail: sherazbintahir@gmail.com

Week 2 – Task 1

SELENIUM SCRAPING

SELENIUM SCRAPING

1. Introduction

Web scraping is the process of extracting data from websites. In this task, we used Selenium (a browser automation tool) to scrape laptop details from Daraz.com. Selenium was chosen because Daraz loads content dynamically (JavaScript-driven), which simple requests + BeautifulSoup cannot fully capture.

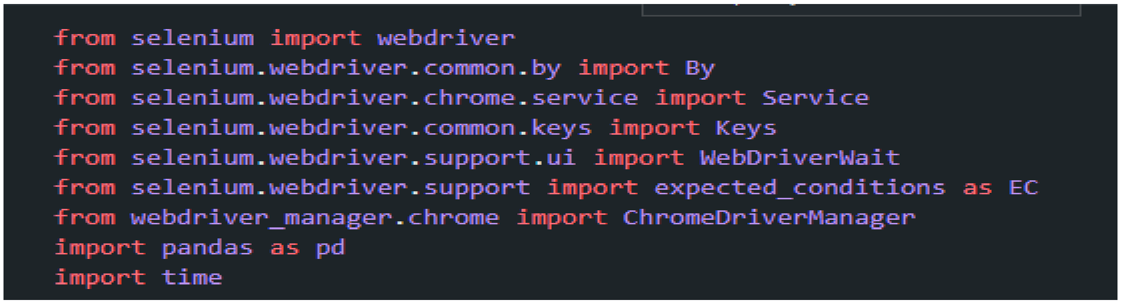
2. Tools & Setup

- Python 3.9 (Conda environment)
- Selenium with ChromeDriver
- Pandas for saving scraped data

3. Scraping Workflow

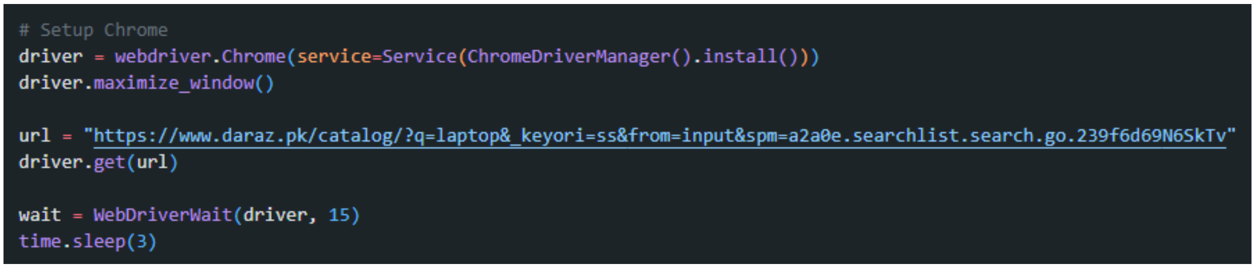
- Launch browser with Selenium.
- Search laptops on Daraz.com.
- Extract details from each product card:
  - Product Title
  - Price
  - Product Image
  - Product Link
- Handle pagination, Navigate through multiple pages.
- Save data into a CSV file.

4. Libraries imported



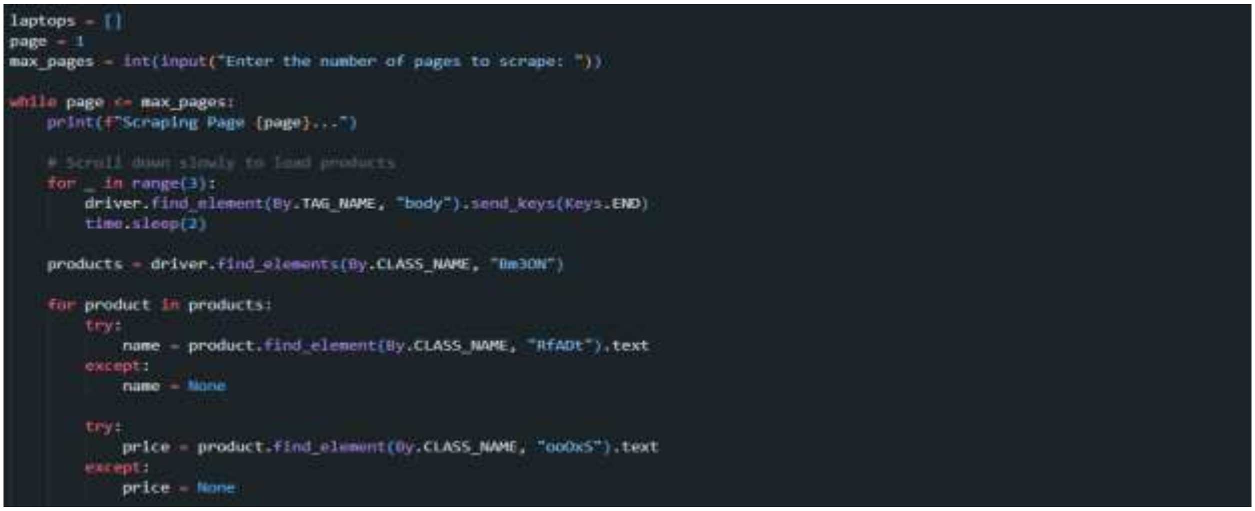
```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager
import pandas as pd
import time
```

5. Chrome Set up

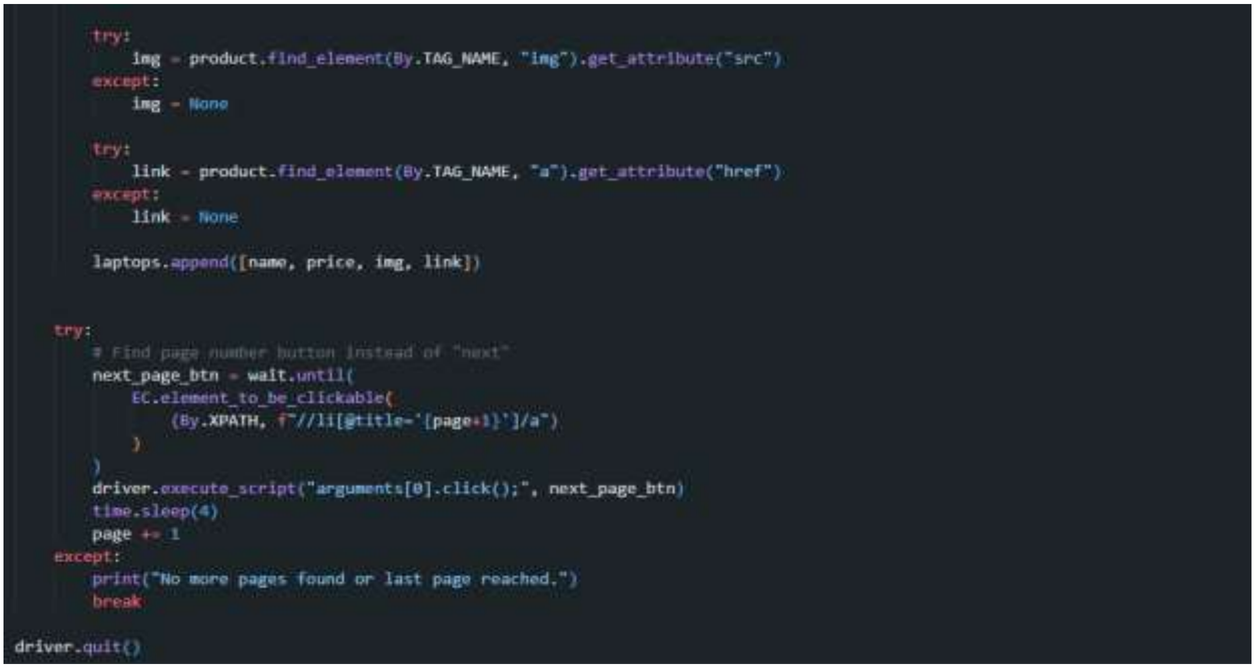


```
# Setup Chrome
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))
driver.maximize_window()

url = "https://www.daraz.pk/catalog/?q=laptop&_keyori=ss&from=input&spm=a2a0e.searchlist.search.go.239f6d69N6SkTv"
driver.get(url)

wait = WebDriverWait(driver, 15)
time.sleep(3)
```

6. Scraping

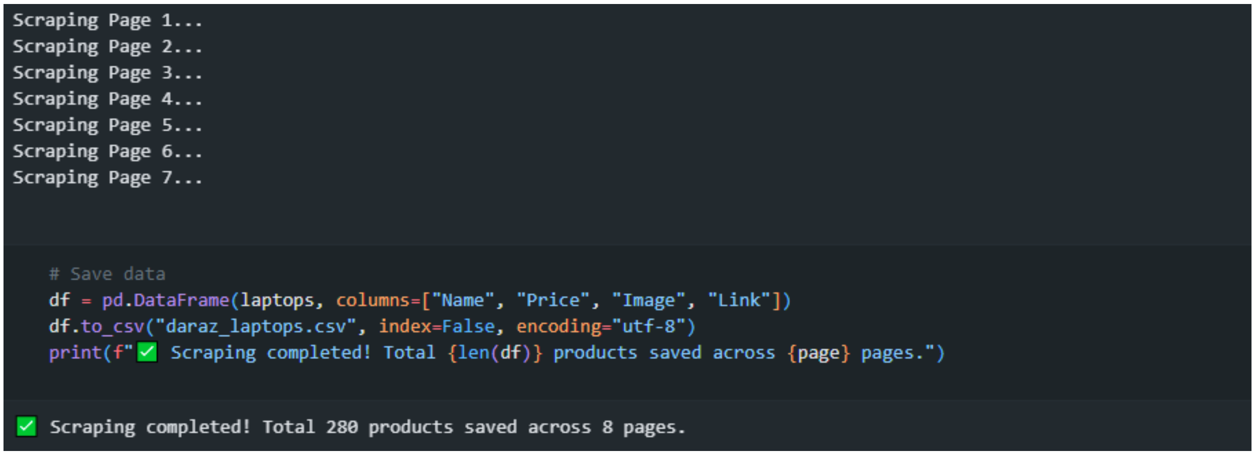


```
laptops = []
page = 1
max_pages = int(input("Enter the number of pages to scrape: "))

while page <= max_pages:
    print(f"Scraping Page {page}...")

    # Scroll down slowly to load products
    for _ in range(3):
        driver.find_element(By.TAG_NAME, "body").send_keys(Keys.END)
        time.sleep(2)

    products = driver.find_elements(By.CLASS_NAME, "Bm3ON")

    for product in products:
        try:
            name = product.find_element(By.CLASS_NAME, "RfADt").text
        except:
            name = None

        try:
            price = product.find_element(By.CLASS_NAME, "ooOxS").text
        except:
            price = None
```



```
        try:
            img = product.find_element(By.TAG_NAME, "img").get_attribute("src")
        except:
            img = None

        try:
            link = product.find_element(By.TAG_NAME, "a").get_attribute("href")
        except:
            link = None

        laptops.append([name, price, img, link])


    try:
        # Find page number button instead of "next"
        next_page_btn = wait.until(
            EC.element_to_be_clickable(
                (By.XPATH, f"//li[@title='{page+1}']/a")
            )
        )
        driver.execute_script("arguments[0].click();", next_page_btn)
        time.sleep(4)
        page += 1
    except:
        print("No more pages found or last page reached.")
        break

driver.quit()
```

7. Saving the CSV file After Scraping pages



```
Scraping Page 1...
Scraping Page 2...
Scraping Page 3...
Scraping Page 4...
Scraping Page 5...
Scraping Page 6...
Scraping Page 7...
```

```python
# Save data
df = pd.DataFrame(laptops, columns=["Name", "Price", "Image", "Link"])
df.to_csv("daraz_laptops.csv", index=False, encoding="utf-8")
print(f"✅ Scraping completed! Total {len(df)} products saved across {page} pages.")
```

```
✅ Scraping completed! Total 280 products saved across 8 pages.
```