Web Scraping of Keyboard Products from Daraz

Submitted By:

Janeeta Ishtiaq

Introduction

In today's digital era, e-commerce websites provide large amounts of product data. This project focuses on **web scraping** product information from **Daraz.pk**, one of the leading e-commerce platforms in Pakistan. The aim is to automatically extract **keyboard product names and prices** using **Selenium**, a Python library for browser automation.

Objectives

- Automate search on Daraz.pk for "Keyboards".
- Scrape product details such as **Title** and **Price**.
- Navigate across multiple pages to collect a wider dataset.
- Save extracted data in **CSV format** for analysis.

Tools and Technologies

- **Programming Language:** Python
- **Libraries Used:**
  - `selenium` (browser automation)
  - `webdriver_manager` (auto driver handling)
  - `pandas` (data storage in CSV)
- **Browser:** Google Chrome
- **Dataset Output:** CSV file (`daraz_keyboards.csv`)

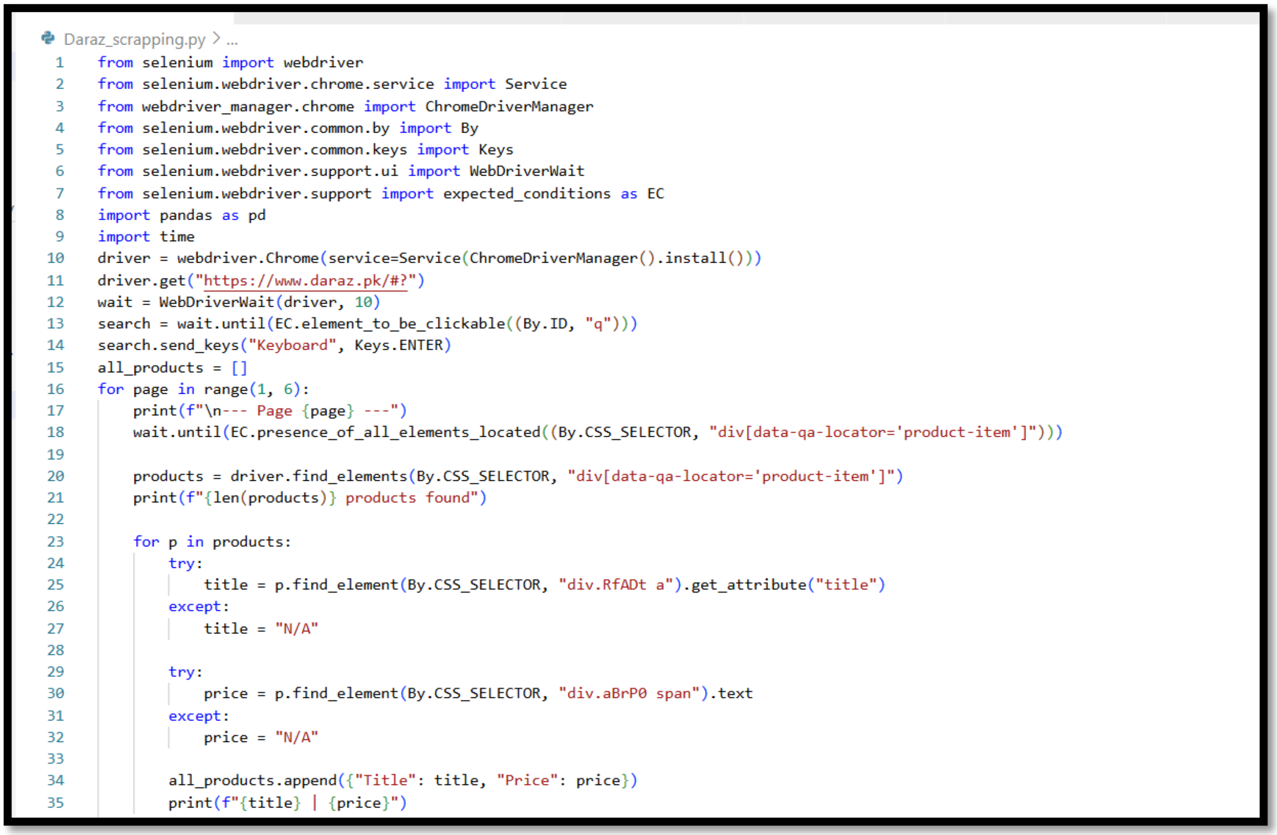
Methodology

The project follows these steps:

1. **Open Daraz website** using Selenium.
2. **Search for "Keyboard"** in the search bar.
3. **Wait for page to load** (using WebDriverWait).
4. **Scrape product data**:
   - Title of the product
   - Price of the product
5. **Loop across multiple pages** (up to 5 pages).

6. **Save results** into a CSV file for further usage.

Code Implementation:



Daraz_scrapping.py > ...

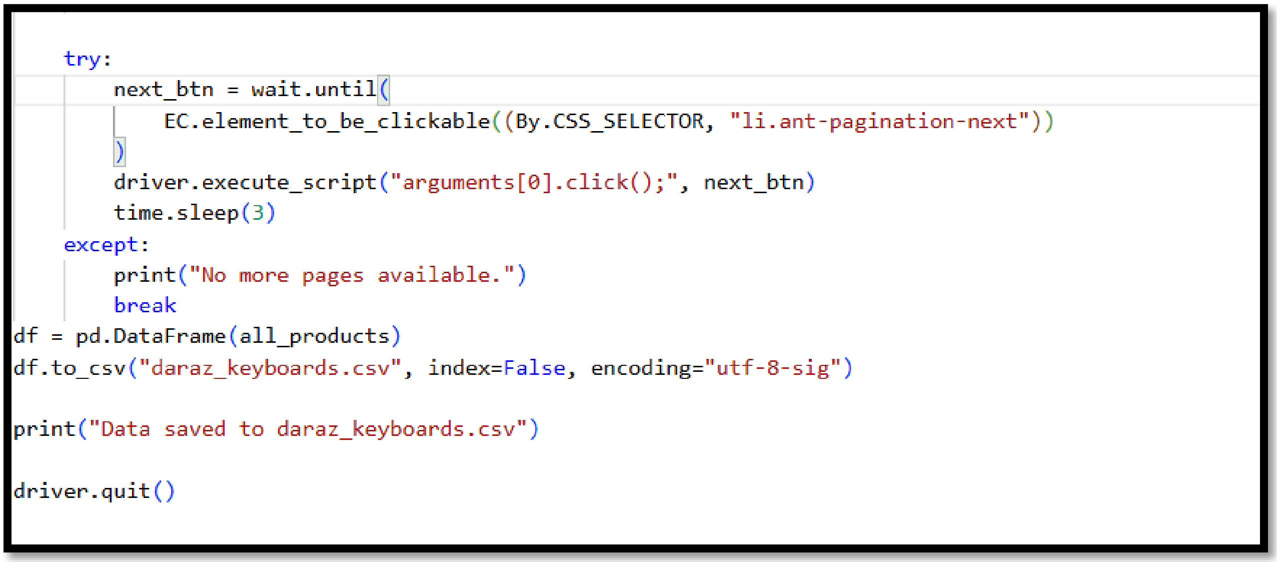
```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import pandas as pd
import time
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))
driver.get("https://www.daraz.pk/#?")
wait = WebDriverWait(driver, 10)
search = wait.until(EC.element_to_be_clickable((By.ID, "q")))
search.send_keys("Keyboard", Keys.ENTER)
all_products = []
for page in range(1, 6):
    print(f"\n--- Page {page} ---")
    wait.until(EC.presence_of_all_elements_located((By.CSS_SELECTOR, "div[data-qa-locator='product-item']")))

    products = driver.find_elements(By.CSS_SELECTOR, "div[data-qa-locator='product-item']")
    print(f"{len(products)} products found")

    for p in products:
        try:
            title = p.find_element(By.CSS_SELECTOR, "div.RfADt a").get_attribute("title")
        except:
            title = "N/A"

        try:
            price = p.find_element(By.CSS_SELECTOR, "div.aBrP0 span").text
        except:
            price = "N/A"

        all_products.append({"Title": title, "Price": price})
        print(f"{title} | {price}")
```

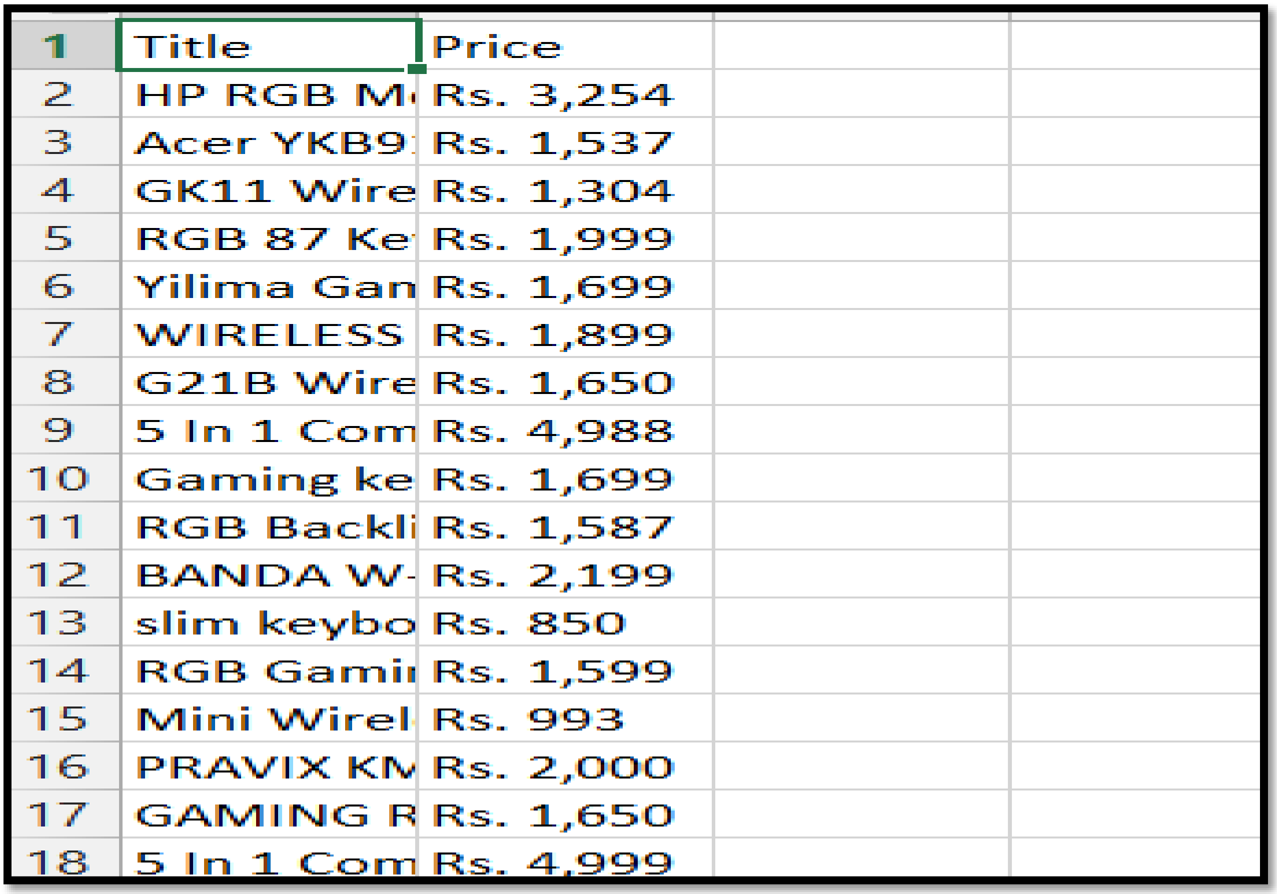


```
    try:
        next_btn = wait.until(
            EC.element_to_be_clickable((By.CSS_SELECTOR, "li.ant-pagination-next"))
        )
        driver.execute_script("arguments[0].click();", next_btn)
        time.sleep(3)
    except:
        print("No more pages available.")
        break
df = pd.DataFrame(all_products)
df.to_csv("daraz_keyboards.csv", index=False, encoding="utf-8-sig")

print("Data saved to daraz_keyboards.csv")

driver.quit()
```

OUTPUT:



1	Title	Price		
2	HP RGB M	Rs. 3,254		
3	Acer YKB9	Rs. 1,537		
4	GK11 Wire	Rs. 1,304		
5	RGB 87 Ke	Rs. 1,999		
6	Yilima Gan	Rs. 1,699		
7	WIRELESS	Rs. 1,899		
8	G21B Wire	Rs. 1,650		
9	5 In 1 Com	Rs. 4,988		
10	Gaming ke	Rs. 1,699		
11	RGB Backli	Rs. 1,587		
12	BANDA W-	Rs. 2,199		
13	slim keybo	Rs. 850		
14	RGB Gamii	Rs. 1,599		
15	Mini Wirel	Rs. 993		
16	PRAVIX KM	Rs. 2,000		
17	GAMING R	Rs. 1,650		
18	5 In 1 Com	Rs. 4,999		

Conclusion

This project successfully demonstrates **web scraping using Selenium**. The program can automatically extract **product titles and prices** from Daraz.pk for multiple pages and save them in a structured CSV file. Such techniques can be extended to scrape other categories and support **e-commerce data analysis**.