



Source - [wikipedia](#)

# Daraz Data Web Scraping Bot Project Report

## Introduction

This project involves creating a web scraping bot designed to extract detailed product information from Daraz, an online shopping platform. The primary objective is to automate the collection of data such as product names, brands, prices, seller details, ratings, and reviews based on specific search keywords. The extracted data is stored in a CSV file for further analysis.

## Objectives

- **Automate Data Collection:** Develop a bot that can automatically navigate the Daraz website, search for products, and extract relevant information.
- **Scalability:** Allow the bot to handle different search keywords and multiple pages of product listings.
- **Data Storage:** Save the scraped data in a structured format, specifically CSV files, named according to the search keyword.

## Tools and Technologies

- **Python:** The core programming language used for this project.
- **Selenium:** A web automation tool used to handle dynamic content and interact with the web page elements.
- **Pandas:** A data manipulation library used for handling and saving the scraped data.
- **CSV:** A format used for storing the scraped data, enabling easy data manipulation and analysis.

# Methodology

## 1. Scraping Workflow

- **Search Keyword Handling:** The search keyword is formatted to be compatible with URL query parameters.
- **Collecting Links:** The bot navigates to the search results page and collects links to individual product pages.
- **Filtering Links:** The bot then filters out certain links based on their positions in the list (e.g., removing every second link).
- **Data Extraction:** For each product link, the bot navigates to the page and extracts data such as product name, brand, price, seller information, ratings, and reviews.
- **Error Handling:** The code is designed to handle scenarios where certain elements are not found on the page, ensuring the bot continues to function smoothly.

## 2. Saving Data

- The extracted data from each product is compiled into a dictionary and then appended to a list.
- This list of dictionaries is then written into a CSV file, where the filename is based on the search keyword, ensuring that the output is easily identifiable.

## 3. Closing Resources

- After all the data is scraped, the Selenium driver is properly closed to free up system resources.

## Results

- **Data Collected:** The bot successfully scraped and collected data for products listed under the search keyword. The data includes:
  - Product Name
  - Brand Name
  - Price before discount
  - Discount
  - Price after discount
  - Seller Information
  - Seller Ratings
  - Product Ratings

- Number of Reviews
  - Questions and Answers
- 
- **CSV Output:** The scraped data is saved in a CSV file named according to the search keyword, which is saved in the working directory. This CSV file can be used for further analysis or as a data source for other applications.

## Challenges and Solutions

- **Dynamic Content:** Some content on Daraz is dynamically loaded using JavaScript. This required the use of Selenium to ensure the bot could interact with and retrieve content that wasn't immediately available in the page source.
- **Error Handling:** The bot needed to gracefully handle scenarios where expected elements were missing, using try-except blocks to manage such cases.
- **Scalability:** Ensuring that the bot could handle multiple pages of product listings and different search keywords required robust looping and string manipulation techniques.

## Conclusion

The Daraz Data Web Scraping Bot successfully automates the process of collecting product information from Daraz based on any given search keyword. The project demonstrates the effective use of web scraping techniques combined with data storage methods, providing a scalable solution for data collection tasks. The extracted data is saved in a structured format, making it readily available for further analysis or use in other applications.

This bot can be further expanded to include more advanced features such as scraping additional product attributes, handling more complex dynamic content, or integrating with a database for storing larger datasets.