

# Web Scraping And Data Analysis For Online Shopping With Selenium

**Lakkakula. Sai Lakshmi<sup>1</sup>, D. Lakshmi Sumithra<sup>2</sup>, K. Jhansi<sup>3</sup>, J. Vandana<sup>4</sup>**

MTech, Computer science & Engineering, Bapatla women's Engineering College, Bapatla, INDIA<sup>1</sup>

BTech, Computer science & Engineering, Bapatla Women's Engineering College, Bapatla, INDIA<sup>2-4</sup>

**Abstract:** In the rapidly evolving landscape of online shopping, the amalgamation of web scraping and data analysis has emerged as a powerful toolset, enabling businesses and data scientists to extract valuable insights and make informed decisions. This paper explores the utilization of Selenium, a robust automation tool, in conjunction with data science methodologies for extracting, processing, and analyzing data from e-commerce websites. The methodology involves the utilization of Selenium, a browser automation tool, to navigate through web pages, simulate user interactions, and extract data elements such as product details, prices, reviews, and other relevant information.

**Keywords:** Web Scraping, Data Analysis, Selenium, E-commerce, Webdriver, Customer behaviour.

## I. INTRODUCTION

Web scraping can be a powerful tool for extracting valuable data about customers, products, people, and stock markets from websites such as e-commerce portals, job portals, and social media channels. This data can be used to gain insights into customer buying patterns, employee attrition behaviour, customer sentiment, and more. The potential applications of web scraping are virtually limitless, making it an indispensable tool for data science professionals.

Using Selenium scripts, we'll collect data from multiple product pages across different online retailers. This may involve handling pagination, dynamic content loading, and other challenges commonly encountered in web scraping.

## II. BACKGROUND & RELATED WORK

The proliferation of e-commerce platforms, the amount of data available for analysis has increased exponentially. Online shopping websites offer a vast array of products, prices, and consumer reviews, making them rich sources of information for market analysis, price comparison, and trend prediction. However, accessing and extracting this data manually can be time-consuming and inefficient. Web scraping is the process of automatically extracting data from websites. It involves fetching the web page, parsing its HTML, and extracting the desired information [1]. In recent years Selenium is a powerful tool for web scraping as it allows for dynamic interaction with web pages, including JavaScript-heavy sites [2].

Research in the field of data analysis for online shopping data covers a wide range of topics, including market segmentation, consumer behaviour analysis, recommendation systems, and pricing strategies [3][4]. Several studies have explored the use of Selenium for web scraping tasks. Research articles and tutorials provide insights into the capabilities of Selenium, its integration with different programming languages (e.g., Python, Java), and its effectiveness in handling dynamic web content [5].

In this project, we aim to leverage Selenium for web scraping to collect data from various online shopping websites. We will focus on extracting product information such as name, price, description, customer ratings, and reviews. Additionally, we will gather data on product categories, brands, and any other relevant metadata.

## III. METHODOLOGY

In the digital age, online shopping has become an integral part of our daily lives, offering convenience and accessibility to a wide range of products and services. With the vast amount of data available on online shopping platforms, there arises a need to extract valuable insights for various purposes such as market research, pricing strategies, and consumer behaviour analysis. In this context, web scraping coupled with data analysis emerges as a powerful tool to gather, process, and derive meaningful insights from online shopping data.

**A. Set up Environment**

Install necessary software and libraries, including Python, Selenium WebDriver, and any additional libraries required for data analysis (e.g., Pandas, Matplotlib).

**B. Web Scraping**

Choose the online shopping websites to scrape based on relevance, popularity, and availability of data. Develop scripts using Selenium WebDriver to scrape relevant data from the selected websites. Extract information such as product names, prices, descriptions, ratings, reviews, and any other relevant data.

Web scraping is the process of extracting data from websites. It involves fetching the HTML content of a webpage and then parsing and extracting the desired information from it. Determine the websites from which you want to scrape data. These could be e-commerce sites, news websites, social media platforms, or any other source of information. Develop the scraping code using your chosen tool or framework.

With Selenium, you would typically write scripts in a programming language like Python. These scripts would automate the browsing process, navigating to the desired pages, and extracting relevant data. Web scraping can be a powerful tool for gathering data from the internet, but it's essential to use it responsibly and ethically, respecting the rights and policies of website owners.

**C. Implementing Selenium**

Selenium is a popular open-source tool used for automating web browsers. It provides a set of tools and libraries for interacting with web elements, simulating user actions, and automating web application testing. It allows users to record interactions with the web application and generates test scripts in various programming languages.

Use Selenium to extract data from web pages by locating specific elements and retrieving their text, attributes, or other properties. Selenium can handle dynamic content loading through techniques like waits and delays to ensure that elements are loaded before interacting with them. Implement error handling to deal with exceptions that may occur during automation, such as element not found or timeout errors.

Selenium WebDriver is the primary component of the Selenium framework that enables automated interactions with web browsers. It provides a programming interface for controlling web browsers and automating actions such as navigating to web pages, interacting with web elements (e.g., clicking buttons, filling forms), and extracting data from web pages.

**D. Data Collection and Preprocessing**

In a web scraping and data analysis project focused on online shopping, you can employ various data analysis techniques to derive insights from the scraped data. Calculate basic statistics such as mean, median, mode, range, and standard deviation of product prices, ratings, or other relevant metrics. Use descriptive statistics to understand the central tendency, variability, and distribution of the data. Compare prices, ratings, or other attributes across different products, brands, or online retailers. Identify outliers or anomalies in the data that may indicate discrepancies or exceptional cases.

**E. Data Analysis**

Exploratory Data Analysis (EDA) is a crucial initial step in the data analysis process, where analysts explore and understand the structure, patterns, and relationships within a dataset. Gather the dataset from relevant sources, which could include databases, spreadsheets, APIs, or web scraping. Clean the dataset to address missing values, outliers, duplicates, and inconsistencies. Examine the basic characteristics of the dataset, such as its size, data types, and variable distributions. Look for recurring patterns or structures in the data that may reveal underlying relationships or clusters. The EDA process provides a foundation for subsequent stages of data analysis, such as predictive modeling, hypothesis testing, and decision-making.

**F. Evaluation and validation**

In the Web Scraping using Selenium project, Evaluation and validation are essential for assess the quality of the scraped data, including completeness, consistency, and relevance. Check for duplicated, missing, or erroneous data and implement data cleaning strategies as needed. By scraping and analysing these diverse types of data, businesses can gain valuable insights into various aspects of their online shopping operations, customer preferences, market trends, and competitor strategies. These insights can then be used to inform decision-making processes, optimize marketing strategies, improve customer experiences, and drive business growth.

## F. Result

At last we get the result of user interface of price checker in that we need to check the prices, reviews, ratings and other information from multiple websites which the data is scraped using selenium. When the data is scraping dynamically from websites the price checker checks the products from different websites and display as a result.

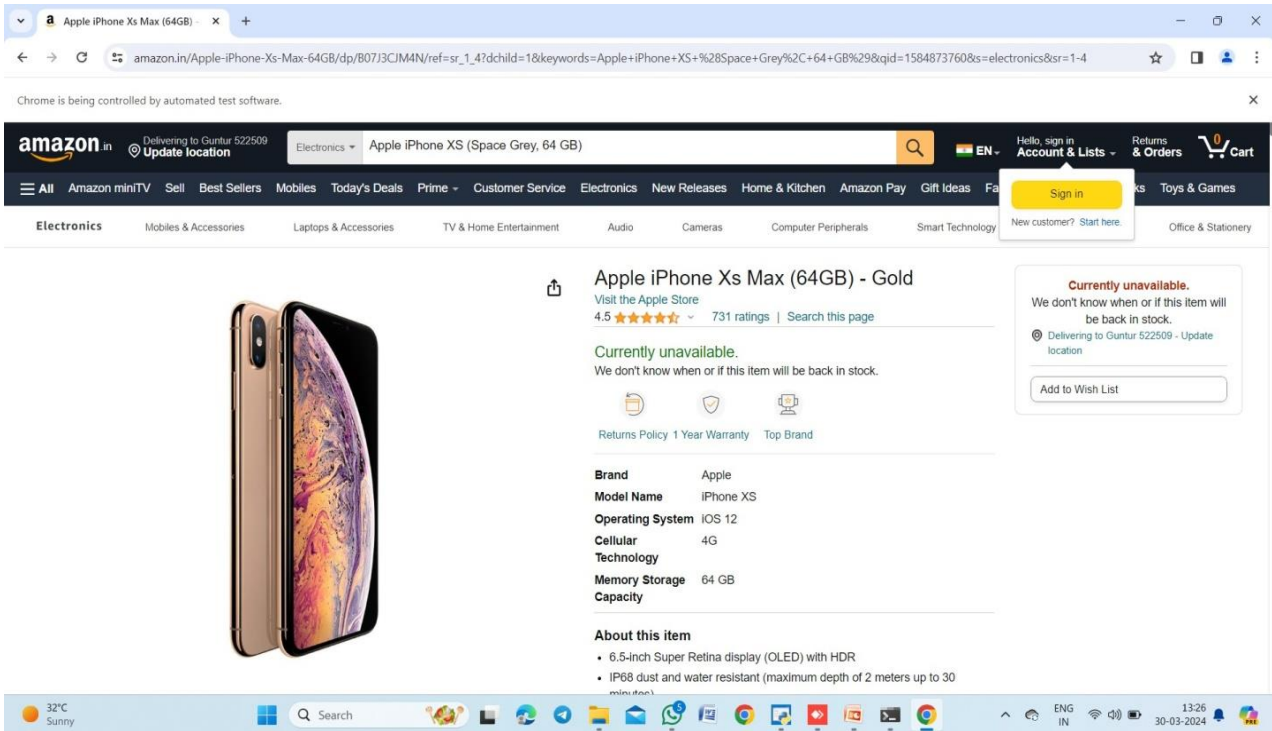


Fig 1: Data scraping from Amazon

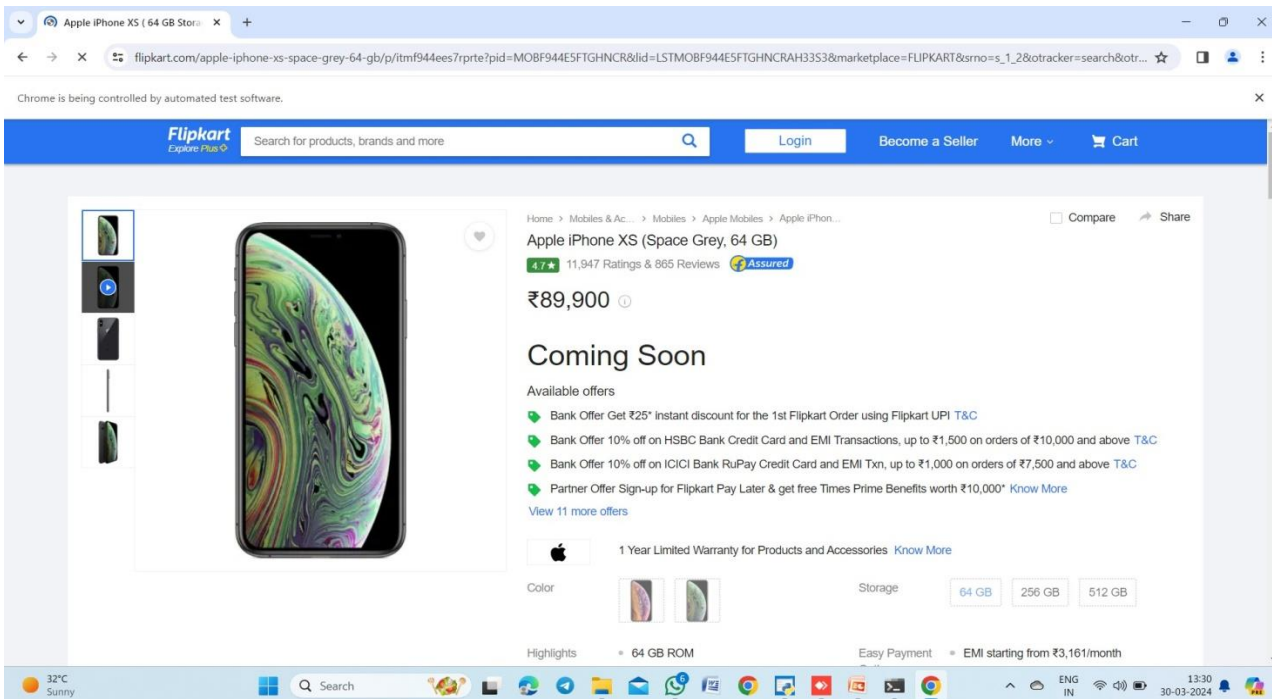


Fig 2: Data Scraping from Flipkart

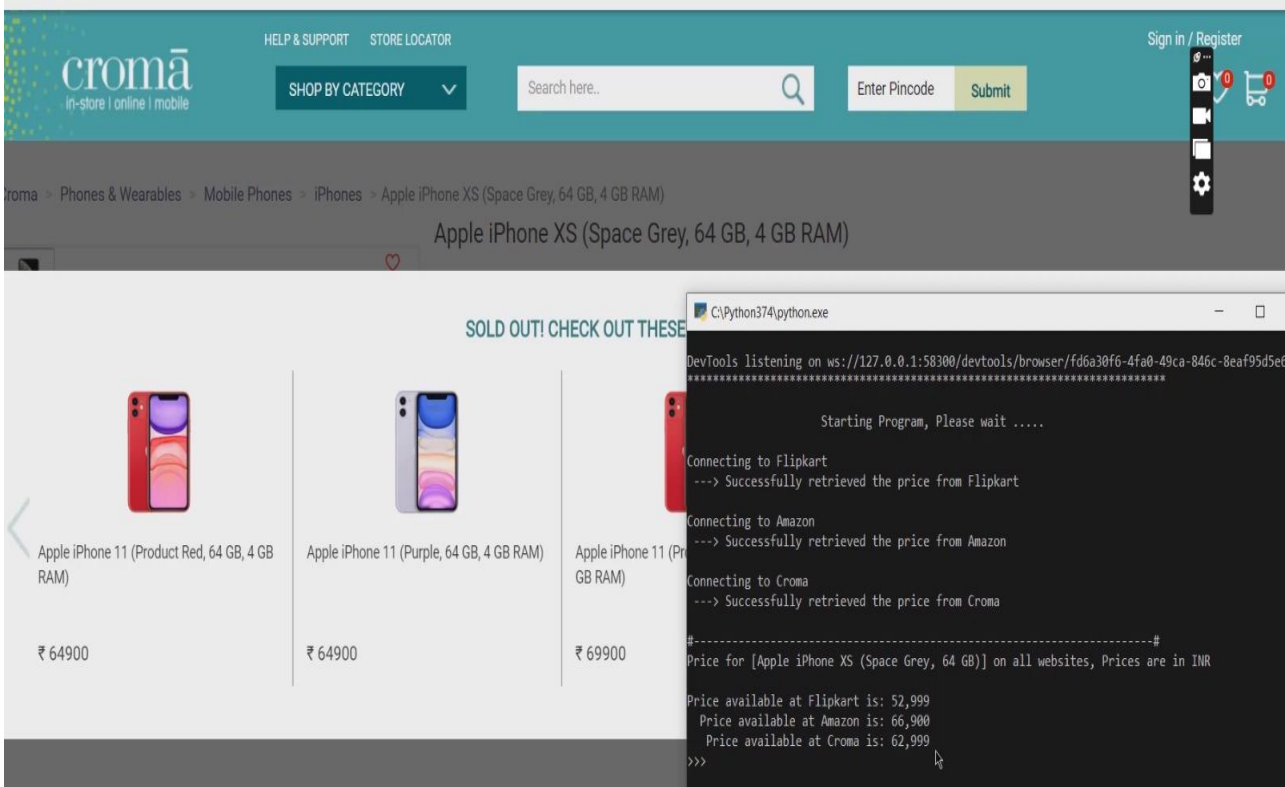


Fig 3: Data Scraping from Cromā

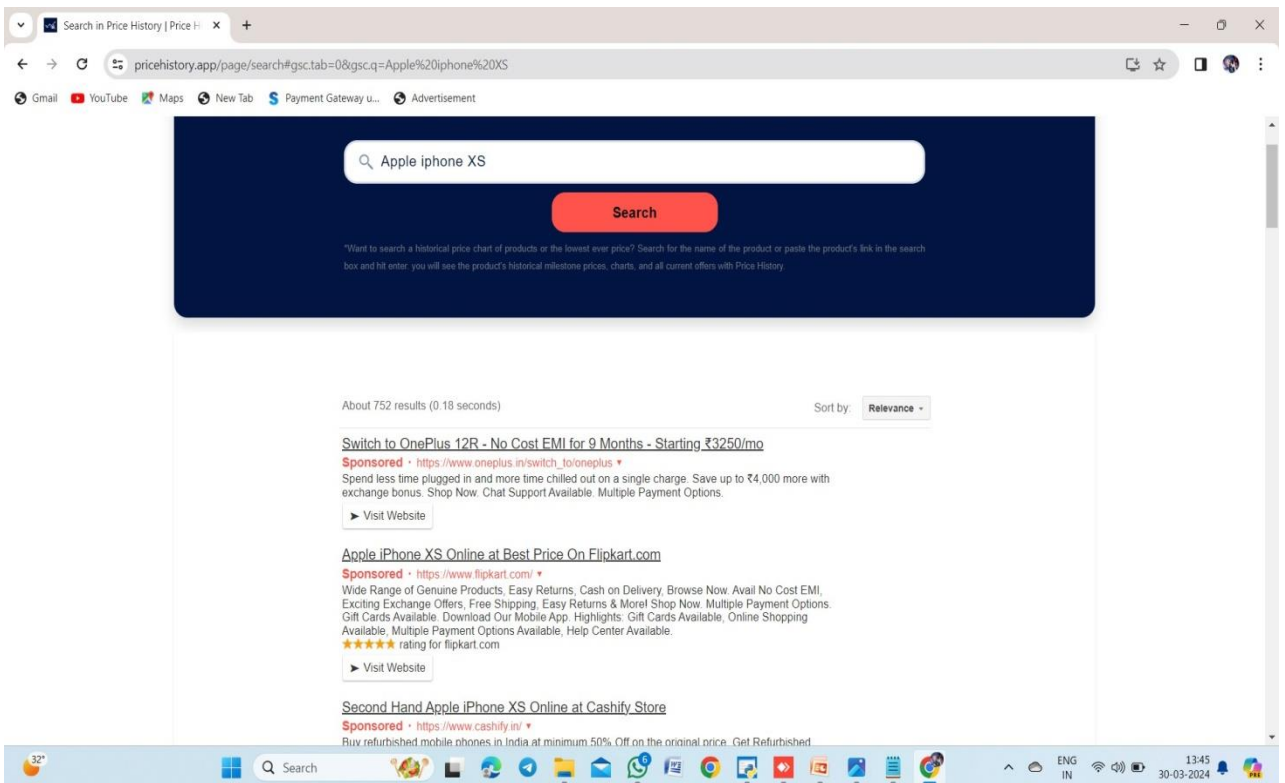


Fig 4: Price checking from websites

**IV. CONCLUSION**

Leveraging the powerful capabilities of Selenium WebDriver, we were able to automate the extraction of valuable data from various online shopping websites. This enabled us to collect comprehensive information such as product details, prices, ratings, and customer reviews, which served as the foundation for our data analysis efforts. Through exploratory data analysis (EDA) techniques, we gained valuable insights into the online shopping landscape. We analysed pricing trends, identified popular products, and assessed customer sentiment toward different products and brands. By harnessing the power of automation and analytics, we have empowered stakeholders to make data-driven decisions and stay competitive in the dynamic world of e-commerce.

**REFERENCES**

- [1]. Beasley, K. (2018). "Practical Web Scraping for Data Science: Best Practices and Examples with Python". O'Reilly Media. Web scraping is the automated process of extracting data from websites. It involves using software tools or libraries to retrieve specific information from web pages, typically in HTML format, and then parsing that data to extract the desired content.
- [2]. F. Farholt, "Less Detectable Web Scraping Techniques," Bachelor Thesis, Linnaeus University, Faculty of Technology, Department of computer science and media technology (CM), 2021. By combining these techniques judiciously, developers can effectively scrape data from websites while overcoming various challenges such as dynamic content, rate limiting, and CAPTCHA challenges. However, it's essential to scrape responsibly and adhere to website terms of service and legal regulations governing web scraping activities.
- [3]. D. M. Thomas and S. Mathur, "Data Analysis by Web Scraping using Python," 2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA). By integrating web scraping with data analysis techniques, businesses can extract valuable insights from online sources to inform decision-making, improve processes, and gain a competitive advantage in various domains such as market research, customer behavior analysis, competitor analysis, and more.
- [4]. L. Richardson, "Beautiful Soup", Crummy, 2020. You can customize the code to extract other types of data or navigate through the HTML structure of the webpage as needed. Additionally, remember to handle exceptions and errors gracefully and to respect the website's terms of service and robots.txt file when scraping data from websites.
- [5]. R. Gunawan, A. Rahmatulloh, I. Darmawan, and F. Firdaus, "Comparison of web scraping techniques: Regular expression, HTML Dom, and xpath," Proceedings of the 2018 International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018), 2019. Beautiful Soup and API access methods are generally faster and more lightweight compared to Scrapy or Selenium for scraping static content, especially when scraping a large number of pages without JavaScript execution or user interaction.
- [6]. Scrap fly, "Web Scraping with Python and BeautifulSoup," Scrap Fly, Jan. 03, 2022. Web scraping with Python and BeautifulSoup is a common approach for extracting data from web pages. BeautifulSoup is a powerful library that makes it easy to parse HTML and XML documents.

**BIBLIOGRAPHY**

**Mrs.L.Sai Lakshmi**  
M.Tech (P.hD), Asst. Professor  
Computer science and Engineering  
Bapatla Women's Engineering College, Bapatla  
Andhra Pradesh, India



**D. Lakshmi Sumithra**  
(B.Tech) Student  
Computer science & Engineering  
Bapatla Women's Engineering College, Bapatla  
Andhra Pradesh, India



**K. Jhansi**

(B.Tech) Student

Computer science and Engineering

Bapatla Women's Engineering College, Bapatla

Andhra Pradesh, India



**J. Vandana**

(B.Tech) Student

Computer science and Engineering

Bapatla Women's Engineering College, Bapatla

Andhra Pradesh, India