**­­**

**PROJECT TOPIC**

**­**

**BY**

**FULL NAME**

**(MATRIC NUMBER)**

**MARCH, 2023**

TABLE OF CONTENTS

**CHAPTER ONE: INTRODUCTION**

[1.1 BACKGROUND INFORMATION 1](#_Toc130613280)

[1.2 STATEMENT OF THE PROBLEM 3](#_Toc130613281)

[1.3 AIM AND OBJECTIVES OF THE STUDY 4](#_Toc130613282)

[1.4 RESEARCH METHODOLOGY 4](#_Toc130613283)

[1.5 SIGNIFICANCE OF THE STUDY 5](#_Toc130613284)

[1.6 SCOPE OF THE STUDY 6](#_Toc130613285)

[1.7 PROJECT OUTLINE 6](#_Toc130613286)

[2.1 INTRODUCTION 8](#_Toc130613287)

[2.2 HISTORY OF WEB SCRAPING 8](#_Toc130613288)

[2.3 TECHNIQUES OF WEB SCRAPING 9](#_Toc130613289)

[2.3.1 Human Copy and Paste 9](#_Toc130613290)

[2.3.2 Regular Expression and Text Gripping 9](#_Toc130613291)

[2.3.3 Http Programming 10](#_Toc130613292)

[2.3.4 Dom Parsing 10](#_Toc130613293)

[2.3.5 HTML Parsing 10](#_Toc130613294)

[2.3.6 Vertical Aggregation Platforms 11](#_Toc130613295)

[2.3.7 Semantic Annotation Reorganizing 11](#_Toc130613296)

[2.4 PRESENT DAY WEB SCRAPING 11](#_Toc130613297)

[2.5 WEB SCRAPING AND E-COMMERCE 12](#_Toc130613298)

[2.6 REVIEW OF EXISTING SYSTEMS 13](#_Toc130613299)

[2.6.1 Pricespy 13](#_Toc130613300)

[2.6.2 Yahoo Shopping 14](#_Toc130613301)

[2.6.3 Shopbot.com 16](#_Toc130613302)

[2.6.4 Tweakers 17](#_Toc130613303)

[2.6.5 Getprice 19](#_Toc130613304)

[2.7 FEATURES OF A WEB SCRAPER TOOL FOR E-COMMERCE STORES 20](#_Toc130613305)

[2.7.1 Intuitive Web Interface 20](#_Toc130613306)

[2.7.2 Ideal Price Comparison 20](#_Toc130613307)

[2.7.3 Advanced Search and Filters 20](#_Toc130613308)

[2.7.4 Availability of Multiple Stores 20](#_Toc130613309)

[2.7.5 Reliable and Authentic 21](#_Toc130613310)

[2.7.6 Save Product Feature 21](#_Toc130613311)

[2.8 DRAWBACKS OF WEB SCRAPER TOOLS FOR E-COMMERCE 21](#_Toc130613312)

[2.8.1 Lack of Trust 21](#_Toc130613313)

[2.8.2 Web Scraping Blockers 21](#_Toc130613314)

[2.8.3 Constant Price Change 22](#_Toc130613315)

[2.8.4 Difficult to Verify Product Quality 22](#_Toc130613316)

[2.9 TERMS ASSOCIATED WITH WEB SCRAPER TOOLS 22](#_Toc130613317)

[3.1 INTRODUCTION 24](#_Toc130613318)

[3.2 REQUIREMENTS ANALYSIS 24](#_Toc130613319)

[3.2.1 Functional Requirements 24](#_Toc130613320)

[3.2.2 Non-functional Requirements 25](#_Toc130613321)

[3.3 SYSTEM ARCHITECTURE 26](#_Toc130613322)

[3.4 SYSTEM DESIGN 26](#_Toc130613323)

[3.4.1 Physical Design 26](#_Toc130613324)

[a) Input Design 27](#_Toc130613325)

[b) Output Design 27](#_Toc130613326)

[3.4.2 Logical Design 27](#_Toc130613327)

[a) Class Diagram 27](#_Toc130613328)

[c) Activity Diagram 30](#_Toc130613329)

[d) Sequence Diagram 32](#_Toc130613330)

[3.5 CONCEPTUAL DESIGN 34](#_Toc130613331)

[3.5.1 Description of Tables 35](#_Toc130613332)

[REFERENCES 37](#_Toc130613333)

**LIST OF TABLES**

**Table Page**

[Table 1.1 Methodology and Tools…………………………………………………………4](#_Toc130161268)

[Table 3.1 Use case narrative for user saving discovered optimal product after comparison………………………………………………………………………………..30](#_Toc130161269)

[Table 3.2 User table………………………………………………………………………36](#_Toc130161270)

[Table 3.3 Product table…………………………………………………………………...37](#_Toc130161271)

**LIST OF FIGURES**

**Figure Page**

[Figure 2.1 Image displaying the web interface of pricespy………………………………14](#_Toc130156789)

[Figure 2.2 Image displaying the web interface of yahoo shopping………………………15](#_Toc130156790)

[Figure 2.3 Image displaying the web interface of Shopbot………………………………16](#_Toc130156791)

[Figure 2.4 Image displaying the web interface of Tweakers.net…………………………17](#_Toc130156792)

[Figure 2.5 Image displaying the shop listing interface of Tweakers.net…………………18](#_Toc130156793)

[Figure 2.6 Image displaying the web interface of Getprice………………………………19](#_Toc130156794)

[Figure 3.1: Class diagram representing a web scraper tool for e-commerce……………..28](#_Toc130156795)

[Figure 3.2: Use-case diagram for web scraper tool for e-commerce……………………..30](#_Toc130156796)

[Figure 3.3: Activity diagram for a user searching a project……………………………...32](#_Toc130156797)

[Figure 3.4: Activity diagram for user saving a product…………………………………..33](#_Toc130156798)

[Figure 3.5: Sequence diagram for a user searching for a product………………………..34](#_Toc130156799)

[Figure 3.6: Sequence diagram for user saving a discovered product…………………….35](#_Toc130156800)

**CHAPTER ONE**

**INTRODUCTION**

# 1.1 BACKGROUND INFORMATION

From the inception of the world to this present day, Trading and commerce has metamorphosed into different forms. The evolution of trading has taken a long walk over the past thousands of years from barter to e-commerce. E-commerce, also known as electronic commerce is simply the buying and selling of products over the internet (Anjali, 2014). Oftentimes, involving the transfer of funds and payments for these products over the internet. As simple as this may sound, e-commerce has revolutionized the global economy of different nations positively.

Before the advent of e-commerce, businesses could only conduct commercial transactions physically thereby limiting the scope of their business since the working hours and customer reach was limited. The emergence of the e-commerce technology has forced many companies to embrace technology and dive into the online market (Silviu, 2010). Big players in the e-commerce industry like Amazon, eBay, and Rakuten generate billions of dollars in annual revenue (Geri, 2022). Statistics even shows that the annual revenue for these mega companies never stays the same as it keeps up-surging every year.

Popular e-commerce Unicorn, Amazon has succeeded in growing its number of employees from 250,000 in 2015 to 1,608,000 in 2021 (Coppola, 2022). For an e-commerce business to strive, It has to put certain factors in place to retain the attention of customers when they visit the online store. One of such factor is the user experience.

Exceptional e-commerce stores standout by ensuring they optimize customer experience to retain customer attention on the online store (Bonastre, 2014). The more time a customer spends on the store, the more products they are likely to purchase. Excellent customer experience ensures navigating the online store to find products on the store is as easy and stress-free as possible. But this is not always enough for customers.

Online shoppers like to take certain measures to ensure they are purchasing products at the most optimal price possible. Research by oberlo.com indicates that more than 81% of online shoppers compare prices from multiple e-commerce stores before making any purchase. This is known as online price comparison.

Online price comparison is the act of comparing the prices and features of products from a number of e-commerce stores in order to make optimal decisions when shopping online. Price comparison is not limited to online shoppers alone. e-commerce platforms compare prices from competitors and use that data to set prices for their own products. This process is called competitive pricing. A scenario where ecommerce stores spy on their competitor’s pricings and decide to sell at a slightly lower price than their competitors to attract customers or sell at a slightly higher price than their competitors to attract higher profits (Blanckenberg, 2022). Competitive pricing is no easy-breezy task as employees will need to manually compare products and prices for thousands of products and then update accordingly on their e-commerce website.

The evolution of price comparison tools will shift the bargaining power to the customers as online stores will engage in competition to present their products at the most optimal price, since customers can now identify price details from multiple websites at a glance. This will lead to an increase in price competition between ecommerce stores, causing price of items to drop significantly. E-commerce stores with prices too high will find it difficult selling to customers (Shin, 2001)

The 21st century has encountered a massive surge in task automation in diverse sectors and industries and the e-commerce sector cannot be left out (Vikram, 2020). Sourcing data for analyzing the e-commerce market and price comparison can be very boring and stressful, and it is for such operations that task automation should be implemented. Data from multiple e-commerce stores can be extracted automatically by a method called web scraping (Cem, 2022).

Web scraping is the automated harvesting or extraction of data from websites. This data can further be used for data analytics and research, replicating web contents and monitoring trends. Price comparison engines automatically extract data from e-commerce stores using web scraping. The data retrieved from these e-commerce stores is then used to compare products price and features for making optimal decisions when shopping online (Kasereka, 2021).

Websites like pricepirates and google shopping have been able to develop price comparison web applications to tackle this challenge but pricepirates have only succeeded in building a web application with a very poor user interface, while google shopping app has been terminated to use google search as its default, making it even more difficult to compare prices. (Ron, 2021) .

# 1.2 STATEMENT OF THE PROBLEM

Research by oberlo.com shows that more than 81% of customers compare product prices from multiple e-commerce websites before proceeding to purchase the product from the most price optimal store.

This comparison between multiple e-commerce stores can be stressful and frustrating as customers will have to shuffle between multiple tabs or e-commerce websites to perform price comparison for products, leading to a bad customer experience and bad customer experience results to lower or even zero purchase.

This project aims to design and build a web scraper tool for e-commerce stores. The web application scrapes product data from multiple e-commerce websites, displays the product and its price to the user in an intuitive web interface for stress-free price comparison between e-commerce stores. The web application also scrapes and displays hot and flash sales from multiple websites in a single page without the need for the user to visit multiple ecommerce stores.

This will eliminate the need to manually switch between tabs of e-commerce websites to compare the price of products hence, delivering better user experience thus leading to more products purchased. A win-win scenario for e-commerce stores and customers accordingly as customers get better shopping experience with ease and the e-commerce stores make more sales effortlessly.

To aid developers, the project further builds an API for developers to easily gain access to its robust functionalities. The API returns scraped data in JSON format for developers to utilize as they deem fit.

# 1.3 AIM AND OBJECTIVES OF THE STUDY

The aim of this project is to design and build a web scraper tool for e-commerce websites to effectively and easily perform price comparison for optimal online shopping

The objectives for the project are as follows:

**Objective 1:** To review and study existing web scrapping tools and price comparison web applications.

**Objective 2:** To design the various components a prototype web scraping tool for scraping data from multiple e-commerce websites.

**Objective 3:** To design the components of an intuitive web application where the scraped information would be organized.

**Objective 4:** To implement and test the web scraping tool and web application.

# 1.4 RESEARCH METHODOLOGY

In order to complete the full-fledged application that will efficiently meet up with all the aim and objectives of this project, the following methodologies were duly followed:

Table 1.1 Methodology and Tools

|  |  |  |
| --- | --- | --- |
| OBJECTIVES | METHODOLOGY | TOOLS |
| **Objective 1:** To review and study already existing web scraper tools and price comparison web applications | In order to carry out an effective research on already existing web scraper tools and price comparison web applications, popular price comparing engines like google shopping, pricespy, intelligencenode, pricebat and pricepirates will be thoroughly studied. What services they offer, How they operate, their functionalities, credits, downsides and limitations. | Google and online price comparison web applications |
| **Objective 2:** To build a web scraping tool for scraping data from multiple e-commerce websites. | Python’s popular web scraping library called beautiful soup was used to build the web scraping tool. | Python and Beautiful soup |
| **Objective 3:** To design and build an intuitive web application for the scraping tool. | To obtain a robust and functional application, the web scraper tool will be integrated into a web application with an intuitive frontend interface built using HTML, CSS and JAVASCRIPT.  The backend of the web application will be built using the Django web framework. | HTML, CSS, Javascript, Python and Django |
| **Objective 4:** To implement, test and deploy the web application. | In order to effectively debug this project, The web application will be tested before deployment. Python’s library for automated testing called unittest will be used to achieve this feat before deploying the web scraping application to pythonanywhere. | Unittest and pythonanywhere |

# 1.5 SIGNIFICANCE OF THE STUDY

This project is highly relevant because it benefits both the online shoppers and e-commerce stores alike. With this project in place, online shoppers can now easily compare prices from their favorite e-commerce stores with just a few clicks without having to switch back and forth between multiple tabs.

The web scraping application rids users of shopping stress and boosts customer shopping experience by getting the best deals available by automating online price comparison (Shalini, 2022). Customers can easily make optimal shopping decisions while e-commerce stores get to make more sales as a result, since impressive customer experience is directly proportional to purchases made.

E-commerce stores can also use the web scraping tool to monitor price fluctuations in their competitor’s online store. They can use that data as a standard for pricing their own product by selling slightly higher to make more profit, or slightly lower to attract more sales.

This project also focuses on building an API with multiple endpoints. With this API, developers can easily integrate the functionalities of the web application into their own application without having to code these features from scratch. Developers can simply make API calls to the web scraper application and get JSON data returned to them. The web application also scrapes and displays all hot deals and flash sales from multiple e-commerce stores, this way, users can view all the hot and recommended deals from their favorite e-commerce stores all in one page without having to visit multiple websites for this data.

The project aims to automate and ease the process of e-commerce price comparison by building a web scraping tool with a topnotch web user interface. The web scraping tool is also focused on delivering better customer shopping experience.

# 1.6 SCOPE OF THE STUDY

1. The web application scrapes data from only three e-commerce websites namely Jumia, Konga and Kaiglo.
2. The web application scrapes only four product categories namely phones and tablets, consumer electronics, fashion wears and computers.
3. The web scraper application is limited to conduct only price comparison for products and not overall category comparison.

# 1.7 PROJECT OUTLINE

Chapter One of the project is the introduction of the project. It contains the background information, statement of the problem, aim and objectives of the study, methodology, significance of the study, limitation of the study and the project outline.

Chapter Two of the project represents the Literature Review. This section mentions and discusses antecedent thesis, research and articles related to the project. The literature review also analyzes pre-existing attempts to solve the problems this project attempts to tackle.

Chapter Three of the project is the system analysis and design. This chapter gives a description of the application design, system requirements and system architecture.

Chapter Four of the project is the system implementation. This chapter explains the modus operandi of the web scraper tool and its components, algorithms, libraries, frameworks, and the user interface.

Chapter Five is the summary, recommendation and conclusion of the project

**CHAPTER TWO**

**LITERATURE REVIEW**

# 2.1 INTRODUCTION

A web scraping tool is a software that is used to scrape, mine or extract data from websites. Collecting data from websites can be done manually by users, but this process becomes really hectic and almost impossible when it involves the extraction of enormous data. Web scrapers help to automate this data collection process from websites by crawling through web pages. Web scraping libraries like beautiful soup scrapes HTML or XML documents and have appropriate python functions available for searching, traversing and modifying the HTML or XML tree to extract the desired data (Zhao, 2017). Companies and businesses need to feed on enormous amount of data to be able to identify trends, gain insight, predict outcomes, study the behavioral pattern of customers, identify business opportunities and strategically position themselves far ahead of their competitors (Majest, 2021).

Before the advent of web scraping, sourcing data has always been a tremendous challenge to businesses that are constantly in dire need of bulky data to gain business insights. Web scrapers have made it easy to spy the websites of competitors in order to make strategic decisions. In this project, we utilize web scraping to extract data from ecommerce websites in order to effectively conduct price comparison for optimal online shopping. This tool will boost the user experience during online shopping as users will not be over worked about making the decision of the most optimal product to purchase. The online comparison application alleviates the stress of choice making by presenting the prices for a given product from different stores in a single intuitive web page.

# 2.2 HISTORY OF WEB SCRAPING

The world wide web was created by Tim Berners Lee in the year 1989 and shortly after, the first web scraper called “The Wanderer” was created by Mathew Gray in June 1993. The Wanderer was created for the sole aim of measuring the size of the web. After the invention of “The wanderer”, scientists were inquisitive and zealous to use this new technology of web scraping to power ideas and create solutions. Hence, it did not take long before the first crawler-based web search engine called Jumpstation was launched that same year.

In the year 2000, salesforce and e-bay launched an API crawler with which developers could gain access to their website’s data and functionalities. This caused a new revolution of web APIs as most other companies followed suit.

Web scraping has taken several forms since its invention. The first form of web scraping was manual copy-and-paste. This was done manually by humans and is not ideal for harvesting big data. Manual copy-and-paste technique of data scraping is very slow and prone to error since the procedure is not automated and is conducted by humans (Khder, 2021). Web scraping with time metamorphosed to include other techniques like regular expressions also known as pattern matching, HTML parsing and DOM parsing.

# 2.3 TECHNIQUES OF WEB SCRAPING

Web scraping can be implemented in a number of ways, each technique with its pros and cons. There are seven major techniques of web scraping that has been widely adopted since the inception of web scraping. These techniques are as follows

## 2.3.1 Human Copy and Paste

This is a manual technique where data to be scraped is copied and pasted by an individual or people. This process involves manually navigating to find relevant data, scraping the data and storing it in any desired data format. This technique is considered to be the first technique of web scraping and some people still do not regard it as web scraping due to the fact that it is not automated. It can be very tasking and takes up a lot of time to scrape significant amount of data. Perhaps, one of the major downsides of this technique is that it is prone to errors and can be almost impractical when scraping enormous data (Raul, 2014).

## 2.3.2 Regular Expression and Text Gripping

This powerful technique uses pattern matching techniques like regular expression and UNIX Gripping to scrape data. Programming languages like Python and Perl are mostly used to execute this technique and it is still widely adopted till this day. Regular expression is a special technique because it does not just scrape or index any data. It scrapes only data that matches the specified regular expression pattern.

Regular expression filters data from the webpage, selects data that matches the specified pattern and then extract the data. This extracted data can then be saved in other popular data formats such as XML, CSV or in a conventional database like SQL. Regular expression patterns can match different text or characters such as digits, non-digits, upper case characters, lower case characters, range of numbers, white space, special characters and many more.

Most languages like Python and Javascript have a specific module for performing regular expressions. Python’s module for regular expression is called the re module.

## 2.3.3 Http Programming

Scraping data from dynamic webpages can be difficult sometimes. Http technique is widely adopted to solve this challenge. HTTP request is sent to the server and a response is received by the browser containing scraped data to be stored in the database or any other data format.

A dynamic web page is simply a webpage that presents different content to different users while maintaining the same interface. Unlike a static web page that displays the same content to all users. Dynamic webpages change frequently making it more difficult to scrape since the HTML structure is not static.

## 2.3.4 Dom Parsing

Modern Browsers like Chrome, Internet Explorer, Mozilla and many others have developer options where the Document Object Model of any webpage can be viewed. Data stored in the DOM tree is then scraped and stored. The DOM is a standard for obtaining, modifying, creating or deleting HTML elements from web pages (Rohmat & Alam, 2018).

Webpages are built using HTML and CSS. The details of websites are hidden within the markup language and can be extracted from the DOM. DOM parsing does more than just access data. DOM parsing is also used to manipulate and alter HTML documents.

## 2.3.5 HTML Parsing

This technique uses query languages like XQuery and HTQL for parsing HTML pages, thereby scraping data from the content of the HTML pages. This technique usually involves consuming and converting unorganized data into structured data that can be easily traversed and used. HTML parsing makes data retrieval easier as data can be accessed in a more structured pattern. XML documents can also be parsed due to its similar syntax to HTML.

HTML parsers process structured and organized markup documents like HTML and XML. Most languages have a specific module to handle HTML parsing. Python’s module for HTML parsing is the HTML Parser module. HTML parsing processes and transforms unstructured data into a format that’s easier to use and read.

## 2.3.6 Vertical Aggregation Platforms

Companies have been able to build custom bots to scrape data from popular websites. These bots have been programmed to self-simulate human activities by monitoring data trends and scraping data. This method has gained popularity due to its ability to extract large data automatically without the need for human intervention. Data retrieved from this technique is properly vetted before storing to ensure validity.

## 2.3.7 Semantic Annotation Reorganizing

Websites that support semantic annotations are scraped using this technique. The annotations are integrated into webpages and then parsed using DOM Parsing. Data can be pulled from any semantic layer of the webpage.

Semantic annotation technique involves tagging web contents with metadata to identify related contents. Tagging web contents semantically makes it easier to find contents on web pages. Scraping data involves finding desired data, then extracting the discovered data. Semantic annotation uses tagging to map contents faster on webpages when traversing and scraping data.

# 2.4 PRESENT DAY WEB SCRAPING

Web scraping has taken a new face over the years. A technology that started as basic manual copy-and-paste has now evolved into an innovation that has powered millions of businesses and startups. The major aim of web scraping is to extract data from websites. As the size of data to be extracted grew to become enormous, human copy-and-paste and basic regular expression could no longer serve for scraping data as this meant massive man-power and time would be required to extract meaningful amount of data.

Present day web scraping has evolved into an automated process. The abundant libraries for web scraping have also contributed positively to this evolution. In this present era, web scraping has proven its relevance in data analysis, marketing, leads generation and price monitoring. Scraped data is usually saved in an excel spreadsheet or CSV while sophisticated scrapers will integrate support for other data formats such as JSON which can then be used in APIs (Sirisuriya, 2015).

More than ever before, e-commerce businesses have been enlightened to understand the relevance of data to business advancement. Hence, businesses invest a lot of resources to obtain big data. To effectively utilize data in business, Big data obtained has to be consumed appropriately to yield maximum business insights and in turn, Business advancement.

In this present dispensation, web scraping APIs have become more popular. Platforms like scrape.do, import.io, scrapy, scrapingdog, octoparse, parsehub and many others have taken the mind of developers as they provide developers with ease and access to a lot of web scraping functionalities without having to write too much complex code (Ozsahan, 2022). Most of these APIs even help overcome web scraping blockers by offering a feature called IP rotation which makes them less likely to be restricted from scraping websites by bot detectors.

# 2.5 WEB SCRAPING AND E-COMMERCE

Web scraping has pushed a lot of businesses into the limelight due to its vast use cases. There are different types of web scraping and each type has its unique role to play in the business sector. It is no longer news that businesses with great infrastructure will do well but businesses that position technology and innovation at the forefront will stay far ahead of its competitors. Web scraping has proven to be a vital strategy for e-commerce businesses to obtain rich data and customer insights. (Kasereka, 2021). E-commerce stores constantly need to spy on competitors to make moves due to the competitive nature of the e-commerce sector. This frequent monitoring can get really difficult and tasking when the frequency and degree of monitoring gets really high. E-commerce businesses cannot efficiently conduct price monitoring manually and this is where automated web scraping comes in to play.

From fundamental economics, The greater the competition, the lesser the price and the higher the quality of products (Doan & Simon, 1996). Web scraping promotes competition amongst e-commerce businesses by simplifying data insights and competitor spying. Thereby leading to a massive cut-down in outrageous pricings in the e-commerce sector. Web scraping does more than just price monitoring. e-commerce businesses can get insight about competitor’s strategy.

# 2.6 REVIEW OF EXISTING SYSTEMS

There are a number of web applications that use web scraping to monitor and compare the prices of products from ecommerce stores. Some of these applications make money by referring customers to online stores after comparison.

## 2.6.1 Pricespy

Pricespy(pricespy.co.uk) is a web and mobile application that utilizes web scraping and data feed to compare prices and products from over 6100 shops. The app has a nice and easy to use interface that makes it easy for customers to navigate and compare products online. Pricespy is absolutely free for customers but charge a certain fee from shops each time a customer is redirected to these online shops from their website. Pricespy updates product price three to five times daily to avoid the shop websites blocking their web crawlers.

The mobile application of pricespy can be downloaded on Google play store. Google playstore is a store where users can download games, apps and digital contents for their android devices. App ratings and reviews are also displayed on the platform to guide users on the best apps to use. Pricespy has ranked greatly on google playstore bagging over 1 million downloads.

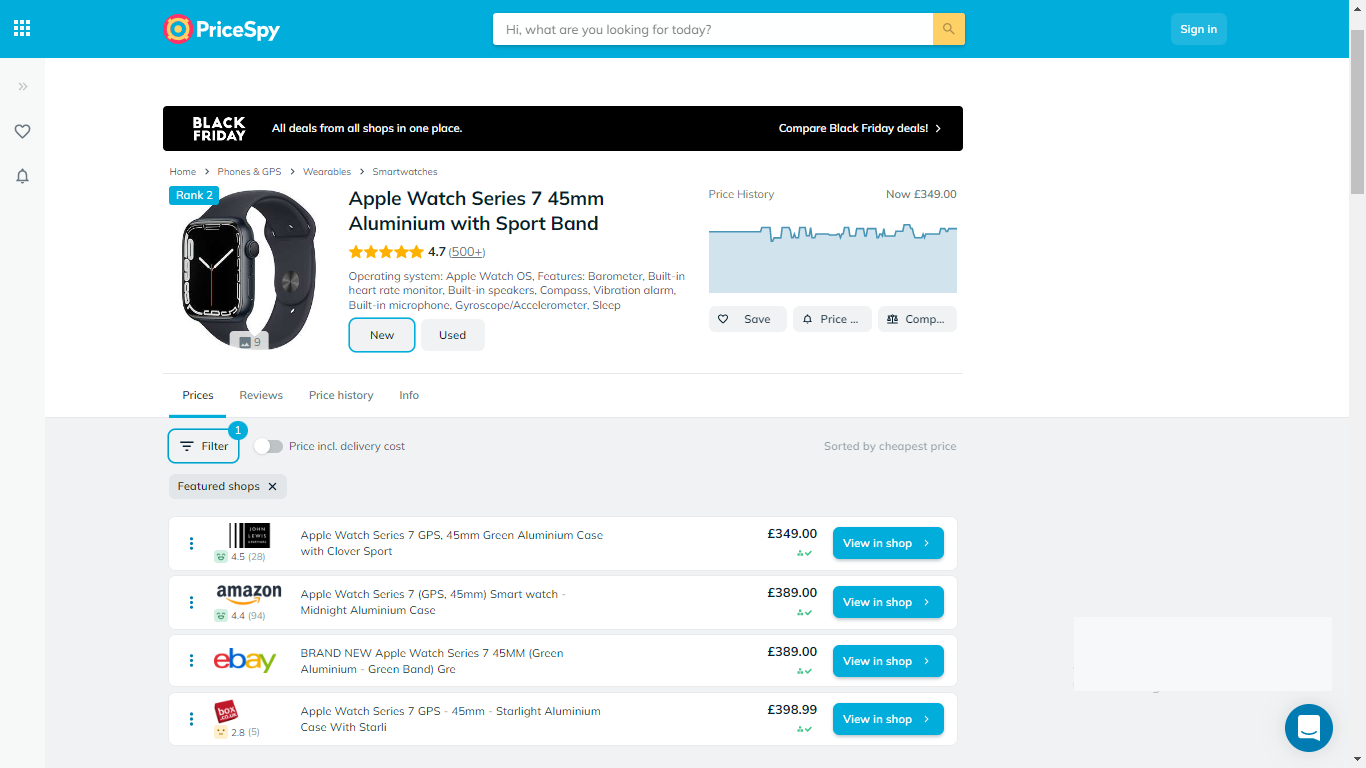


Figure 2.1 Image displaying the web interface of Pricespy

(Source: pricespy.co.uk)

Figure 2.1 above shows the web user interface of pricespy. The web application presents products in a horizontal list with buttons to redirect users to the merchant store for purchase completion.

Some of the features of Pricespy include:

1. **Multiple Shops:** Compare product prices across 6100 shops. This way customers can make optimal shopping decisions
2. Product reviews and comments
3. **Price History:** View the price trend in a graphical format. This feature will show users how the price for a particular product has changed over time
4. **Save Product:** Discovered products can be saved and processed later.
5. **Multilingual:** Website can be viewed in English, French, Norwegian, Danish, Finish and others.
6. **Price Notification:** Users get notified when the product price drops to your desired price.

## 2.6.2 Yahoo Shopping

Yahoo shopping is an online comparison tool that compares price, track price changes and scrapes deals from online shops for optimal shopping. Yahoo shopping redirects customers to merchant websites but earn a small fee from these referral links. This platform displays products from a wide variety of shops and product categories to promote the satisfaction of customers.

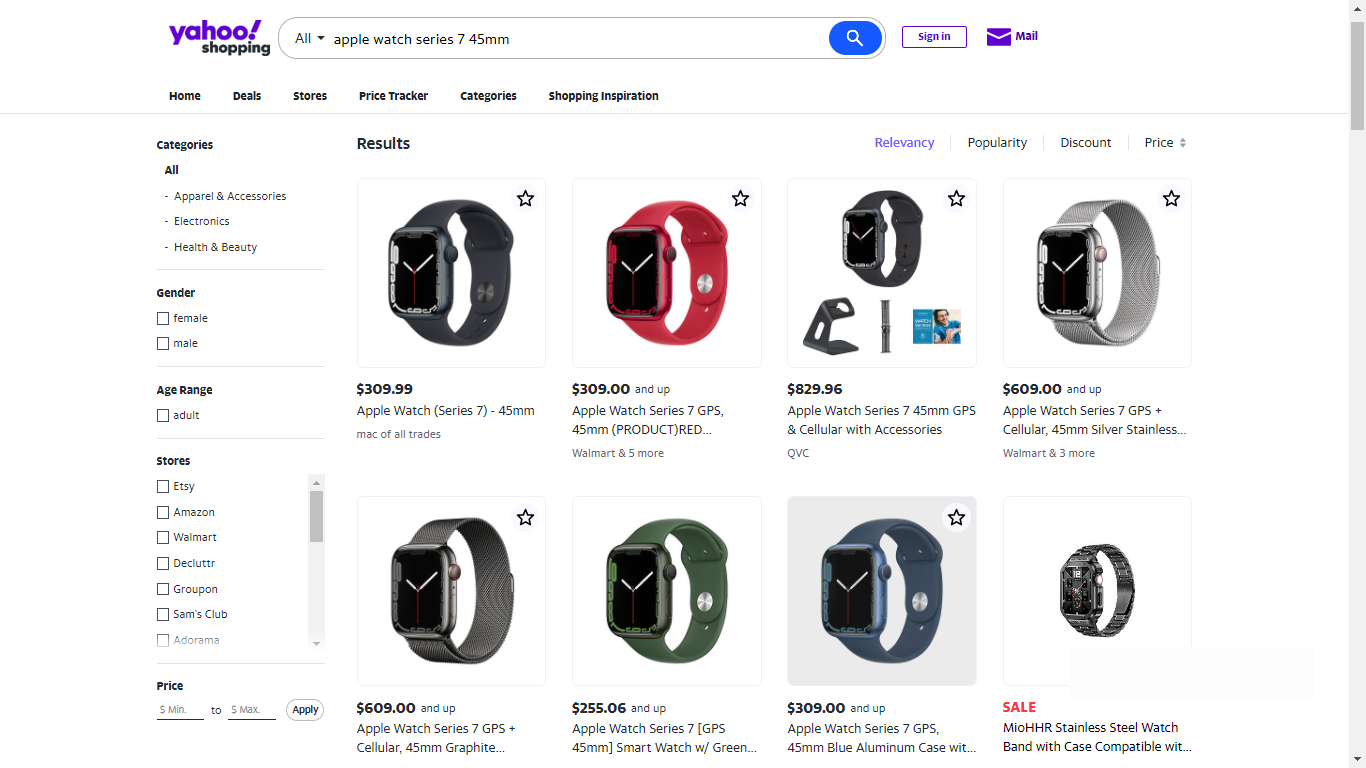


Figure 2.2 Image displaying the web interface of yahoo shopping

(Source: shopping.yahoo.com)

Yahoo shopping displays searched products in a slightly different manner. As shown in Figure 2.2, products are presented in a grid format as opposed to vertical listing of products as seen in pricespy.

Some of the features of yahoo shopping include:

1. **Multiple Shops:** Compare product prices across multiple online shops. This way customers can make optimal shopping decisions.
2. **Price Tracker:** This feature tracks the price of products added to wish-list and sends an alert when the price drops to the user’s affordable rate.
3. **Advanced search filter:** Product search can be trimmed down by setting criteria for products to be indexed such as maximum and minimum price. When a price filter is applied, only products within that price range will be displayed to the user.

## 2.6.3 Shopbot.com

Shopbot is an online comparison engine that uses web scraping to perform price analysis and compare products based on their price, reviews and features. Shop bot is an Australian company that conducts product comparison across products of diverse categories. Prices on the platform are updated daily but the final price on the merchant’s website could sometimes be slightly different from that on the comparison website as price updating is not implemented frequently.

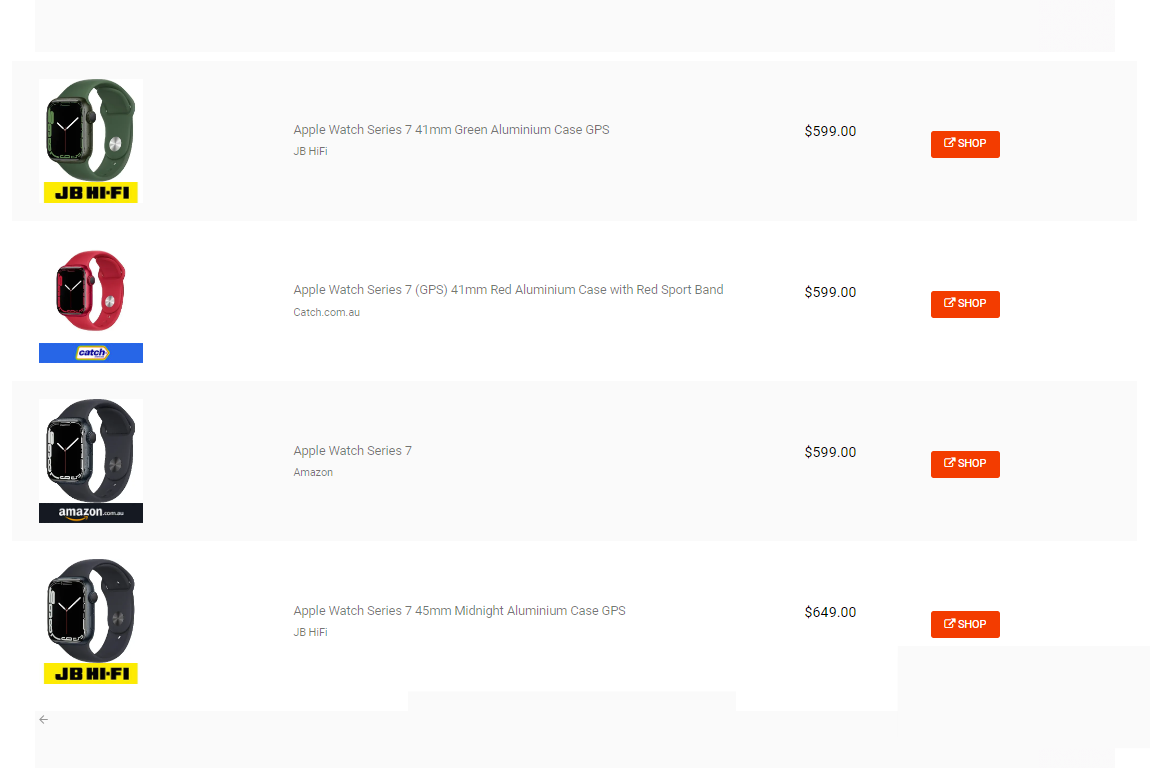


Figure 2.3 Image displaying the web interface of Shopbot

(Source: shopbot.com.au)

Some of the features of Shopbot include:

1. **Vast Product Categories:** Availability of vast product categories. The platforms cover electronics, gadgets, kid wears, health and beauty products and others.
2. **Simple Interface:** Intuitive web interface for easy navigation.
3. **Awesome User Experience:** Price comparison in just few clicks.

## 2.6.4 Tweakers

Tweakers is a Dutch website for comparing products, prices and shops. The website is multi-functional because it compares products, compare prices and also covers tech news on the homepage. Tweakers covers a lot of product categories.

The website tracks the location of location of visitors and offers website language options using Google translate. The two languages supported on the website are English and Dutch. When comparing product price, the website also displays reviews on the product been searched. These reviews can be user reviews in text format, or company reviews in video format.

Tweakers was built by a computer scientist called Femme Taken as a student’s hobby project in the year 1998. The technology has grown to become popular in Netherland and Belgium, winning multiple awards over the years. Currently, the web application has ammased over 1 million members, with its forum having more than 29 million posts. Tweakers was awarded website of the year, six times in a row starting from 2009.

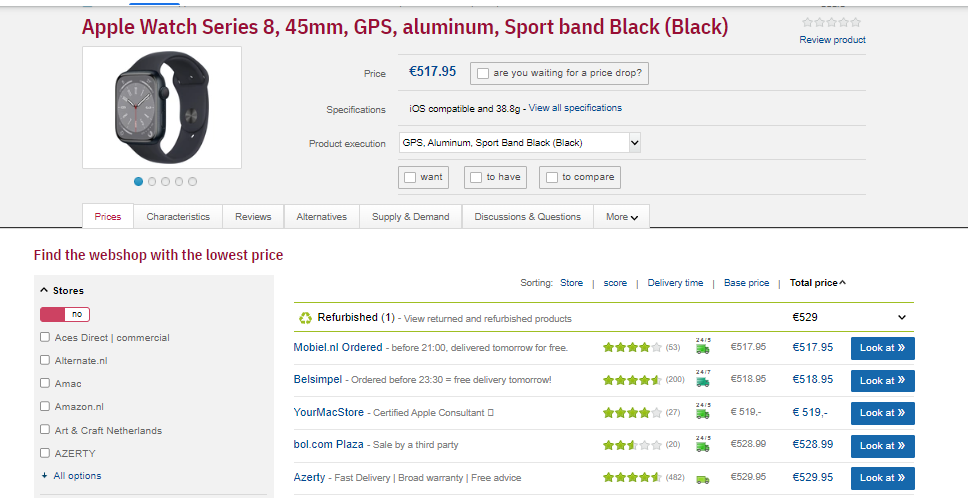


Figure 2.4 Image displaying the web interface of Tweakers.net

(Source: tweakers.net)

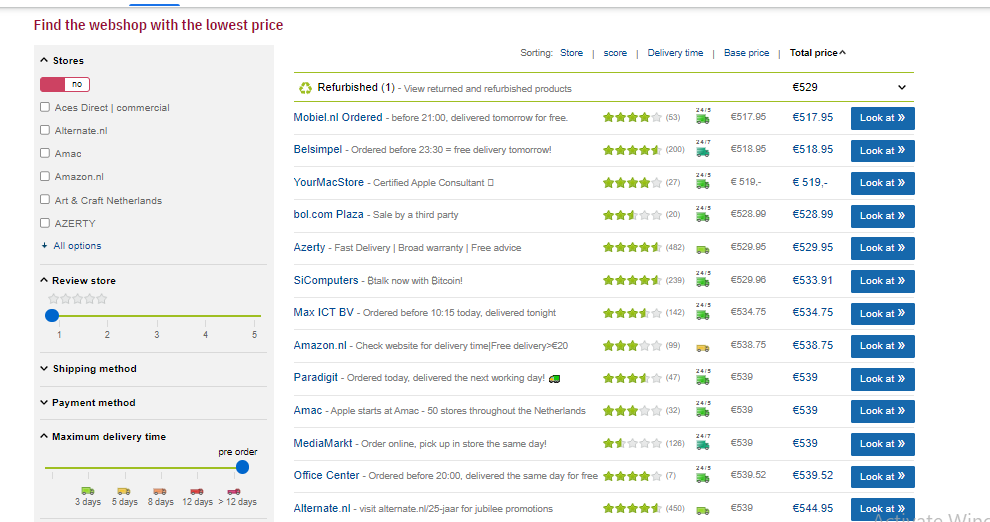


Figure 2.5 Image displaying the shop listing interface of Tweakers.net

(Source: tweakers.net)

Some of the features of Tweakers include:

1. **Multiple Shops:** Compare product prices across multiple online shops. This way customers can make optimal shopping decisions.
2. **General Product Review:** Tweakers display user and company reviews. These reviews have the capacity to influence the choice of buyers by pointing out the characteristics and experiences with the products. Users can also rate products by picking any rating from 1 to 5 star for the product.
3. **News Aggregator:** For each product searched, Tweakers displays all news stories related to the product searched. Thus, Tweakers functions both as a news aggregator and online price comparison tool.
4. **Advanced search filter:** Product search can be trimmed down by setting criteria for products to be indexed such as maximum and minimum price. When a price filter is applied, only products within that price range will be displayed to the user. Products can also be filter via star rating such that only products with 5-star rating will be displayed to the user.

## 2.6.5 Getprice

Getprice is an online price comparison website for finding the best deals in Australia. The website covers products from diverse categories such as electronics, computers, clothing and fashion, sports and traveling, health and beauty, kids and toys, office supplies, cars, accessories and many more.

Getprice also has a section for displaying news stories on products, ecommerce, sales and technology. The online comparison website compares products from over 360 stores. Getprice lists over 3.5 million products from more than 1200 retail outlets in Australia, pulling a monthly traffic of more than 1.5 million users.

The web application stands out due to its modest user interface, responsive design and intuitive layout. Making it easy for users to navigate and find products.

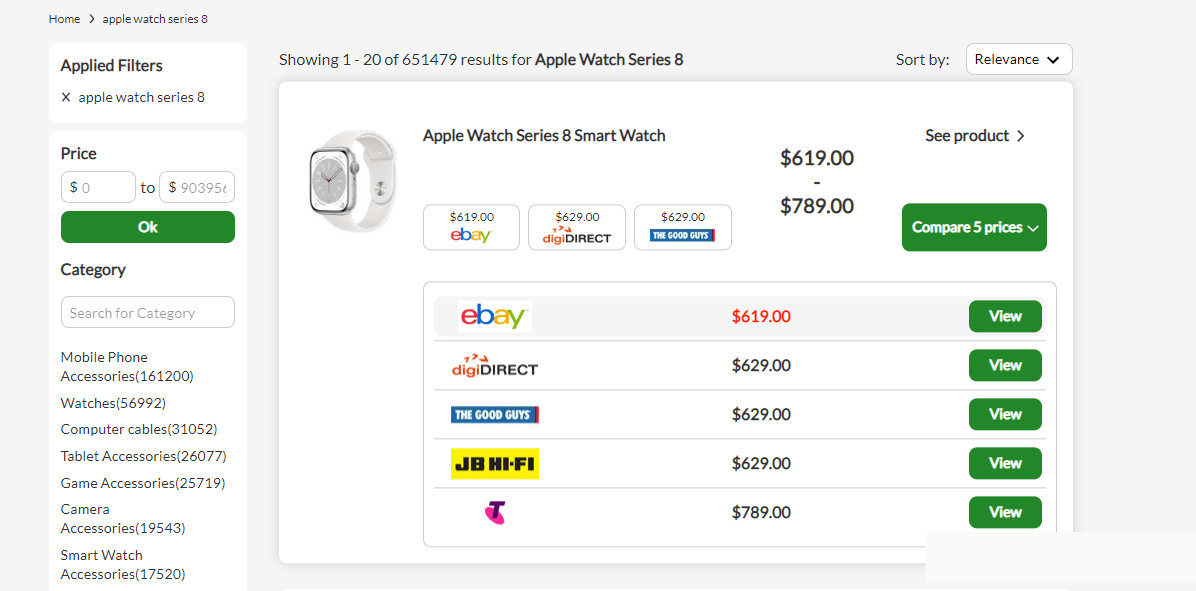
****

Figure 2.6 Image displaying the web interface of Getprice

(Source: getprice.com.au)

Some of the features of Getprice include:

1. **Multiple Shops:** due to the availability of vast shops in the web app, product prices can be compared across over 350 online shops. This way customers can make optimal shopping decisions.
2. **Advanced search filter:** Product search can be trimmed down by setting criteria for products to be indexed such as maximum and minimum price.
3. **Intuitive Layout and Interface:** Intuitive web interface for easy navigation. Making it easy for users to find optimal products

# 2.7 FEATURES OF A WEB SCRAPER TOOL FOR E-COMMERCE STORES

Web scraper tools for e-commerce businesses are built to satisfy the needs of online shoppers. For a web scraper tool to be exceptionally effective, it must possess certain vital features to be able to deliver top-tier service and standout from similar web applications.

## 2.7.1 Intuitive Web Interface

An intuitive web interface is an interface that is predictable and easy to understand. A web scraper tool should be easy to use without requiring the user to overthink. The more intuitive a web interface, the less confused a user is likely to get when using the web application. Bad web interface might require the user to go through some sort of tutorial to be able to understand and use the web application and this can be a turn-off to new customers.

## 2.7.2 Ideal Price Comparison

An ideal price comparison tool is one that is focused on satisfying customers by meeting the demands for which the customers visit the web application. Customers use comparison tools in order to find the best product as the most optimal price possible. The customer should get nothing short of this. A satisfied customer is definitely bound to return to complete more deals.

## 2.7.3 Advanced Search and Filters

The ability to navigate and find desired products in the web application should not be difficult. Users should be able to narrow down their search based on certain criteria such as maximum and minimum price, discount percentage and shops. This saves time and resources as users will not have to navigate much before finding the product of their choice.

## 2.7.4 Availability of Multiple Stores

Customers use online comparison applications because they want to compare product prices across multiple shops before making purchase. The aim is defeated when there are no multiple credible shops to compare products across. Vetting shops before adding them to be comparison list will help build customer trust.

## 2.7.5 Reliable and Authentic

Product price on comparison websites should correspond with the price in the merchant’s website. Customers will feel frustrated if they notice a cheap price in the comparison website, only to find a more expensive price when they proceed to complete the purchase on the merchant's website. There should be a high degree of synchrony between prices on online comparison websites and merchant websites.

## 2.7.6 Save Product Feature

Customers should be able to save discovered products for later use. More like a cart feature where products that customers have indicated interest in will be saved for future processing. This feature will avoid customers searching repeatedly for the same products thereby saving time and resources.

# 2.8 DRAWBACKS OF WEB SCRAPER TOOLS FOR E-COMMERCE

The past two decades have recorded an exponential adoption of web scraper tools for e-commerce. But this advancement has encountered a bit of limitation due to certain unfavorable factors. Some of the drawbacks of web scraper tools are outlined below.

## 2.8.1 Lack of Trust

Customer participation can dwindle if there is a noticeable absence of trust when shopping online. This lack of trust can be attributed to a number of factors ranging from uncertainty, bad past experience, poor user interface and experience, unsatisfied customer expectations and inconsistent price between merchants and comparison websites (Lanford, 2006). One-way web scraper tools can curb this problem is to list only vetted online stores who have a track record of delivering quality products and top-tier products to customers.

## 2.8.2 Web Scraping Blockers

Most modern websites use certain techniques to block web scrapers from accessing the contents of their websites. This makes it extremely difficult to extract data from such websites. Some of such techniques include User behavior analysis such as CAPTCHA, IP address-based blocking, and IP rate limiting. These methods are usually put in place to differentiate human users from bots.

Some websites can monitor the rate of requests sent within a given time. The system will assume the user is a bot if the number of requests made is too high as only bots can send many requests more than a human can possibly generate within a short time frame.

## 2.8.3 Constant Price Change

Most web scraper tools for e-commerce like Pricespy update their price three to five times daily. Thus, price change will not be effected on the online web scraper tool if the change is effected. Advanced bot detectors will tend to block web scrapers if unlimited requests are made to the website been spied.

## 2.8.4 Difficult to Verify Product Quality

Ecommerce web scraper tools are simply a connecting link between the customers and online shops. These platforms are not the merchants in reality. Hence, they cannot totally vouch for the quality of the products they recommend to you. Customers who find optimal products through an online comparison website tend to blame online comparison tools for faulty purchased items. Failing to understand that these platforms are simply connecting links to the original merchant stores. As a result, most modern web applications like price spy and yahoo shopping have terms and condition embedded on their websites to help customers better understand the terms and conditions binding the services rendered. Customer satisfaction is dependent on service quality and delivery (Brown & Jayakody, 2008).

# 2.9 TERMS ASSOCIATED WITH WEB SCRAPER TOOLS

Below are some terms used and their definitions.

1. **API**

API is the acronym for Application Programming Interface which represents a software that helps two applications communicate and interact.

1. **Bot**

A bot is a software or program created to simulate human activity. Bots are automated to perform tasks without human intervention.

1. **XML**

XML is the acronym for Extensible Markup Language. This is a set of codes and tags that are used to store and share structured data across systems and networks.

1. **JSON**

JSON stands for JavaScript Object Notation and it is a common text-based data format for storing and sharing data comprising of collections of key-value pairs.

1. **Feed**

A program that provides users or websites with new and updated contents from a source application

1. **Interface**

Interface is the element of interaction between users and computers, making it possible for users to send input to and receive output from the application.

1. **Libraries**

A Library is a collection of closely connected modules bundled together to help programmers easily implement tasks.

1. **CSV**

A CSV (comma-separated values) is a text file for holding data separated by commas where each line of the file represents a data record allowing data to be stored in a table structured format.

1. **Parsing**

Parsing is the process of converting a code or program into a format that a runtime environment can execute or run.

**CHAPTER THREE**

**SYSTEM ANALYSIS AND DESIGN**

# 3.1 INTRODUCTION

This chapter covers the analysis and design phase of the system. System analysis depicts where the system will be used, for what purpose the system will be used, the feasible features within the system, and the organisational structure. Here, we also examine the system requirements and map them out as a guide to building robust software.

The system design, system architecture, and all UML diagrams necessary to better comprehend the workings and operations of the web application will be illustrated in this chapter. UML diagrams covered include the use case diagram, activity diagram, sequence diagram, class diagrams, and a few others.

System analysis and design aim at understanding what humans or users require to be able to analyze systematic data flow and inputs, application processes, operation flow, and output information of a system

## 3.2 REQUIREMENTS ANALYSIS

System requirements are the structure and configuration that a system must possess for it to perform at an efficient rate. The process of software development involves requirements gathering. Requirements gathering is the process of figuring out the exact requirements of the software or project from inception to completion. This involves a thorough understanding of the system requirements of the software application (Kendall, 2014).

System requirements can be classified into two categories namely: functional and non-functional requirements. These requirements go hand in hand and should be treated together as highly relevant requirements (Salinesi & Regnell, 2002).

### 3.2.1 Functional Requirements

Functional requirements are the requirements that describe what a software application should do and how it should behave. Just as the name implies, functional requirements make clear the functions of a system. These functional requirements are compulsory and must be met for the system to be termed a functional or working system It is very necessary to outline the functional requirements of the system so that users can identify what the system can and cannot perform. These requirements will map out the functional and operational boundaries of the system. Below are the functional requirements of the Web scraper tool for e-commerce stores:

1. The user shall be able to search for products
2. The user shall be able to compare products over multiple online stores
3. The user shall be able to save products
4. The user shall be able to display hot deals from multiple online stores
5. The user shall be able to sign up or register to the system
6. The user shall be able to log in and logout of the system

### 3.2.2 Non-functional Requirements

Non-functional requirements are the requirements that drive quality in service delivered to users. Non-functional requirements are requirements that describe how the system works and are not compulsory to implement the functionalities of the system but can determine the quality of service rendered. Some of the non-functional requirements include:

1. Usability requirement: The system should have an intuitive user interface such that users can easily understand. An intuitive user interface is an interface that requires less thinking or attention to understand. Users should be able to figure things out on their own when using the system.
2. Performance requirement: The system should run efficiently without glitches and errors. The system should be able to accomplish and complete tasks within the shortest possible time to avoid a bad user experience. The response time of the system should be minimal such that users do not have to wait for too long to get feedback and desired results.
3. Reliability requirement: The system should only accommodate minimal errors such that the system can still function manageably. At no point in time should major or every component of the system be faulty all at once to eliminate the possibility of a total system failure.
4. Security requirement: The system should be able to resist malicious attacks. The system should provide confidentiality, availability, and integrity security. User data should be kept confidential and safe. The website should be resilient enough to resist attacks from malicious sources.

# 3.3 SYSTEM ARCHITECTURE

A system is made up of internal components incorporated together to operate as a whole. System architecture depicts the structure of the components that make up the entire system or application (Gorton, 2011). Some of the components of the system include the Scraping module, Comparison module, Save module, recommendation module, and API module.

The scraping module handles the scraping of data from multiple e-commerce websites. This module is the engine behind obtaining useful e-commerce data. The comparison module presents the data obtained from the scraping module in a comprehensible interface for easy comparison such that users can compare product prices and reviews at a glance. Products that have been selected by the user can be saved for further processing. The save module handles saving desired products to the database. Products and hot deals will be handled by the recommendation module.

The API module is responsible for the interface that helps developers and other software communicate with the web scraper tool. With this module, independent developers can send requests through the API and get feedback data that can then be further utilized and used as they desire.

# 3.4 SYSTEM DESIGN

System design entails defining the components, modules, and interfaces of a system based on some specified system requirements. System design has three major specifications namely Physical design, Logical design, and Conceptual design.

## 3.4.1 Physical Design

Physical design depicts the exact input and output processes of the system. This shows the life-cycle of data obtained from users, down to the output displayed. Physical design is responsible for showing how data is received from the user, verified, processed, and presented to the front end as output. Physical design can further be divided into two categories namely: Input design and output design.

### a) Input Design

Input structures are the data entry point into the system and they partly determine the quality of output or result the system will deliver. The input data can be any form of data received from the user such as form data, product search data, user information and user feedback. Effectively designed input system comprises of screens and forms for receiving user data.

### b) Output Design

The output mechanism is responsible for presenting results appropriately to users The user interface for display output to the users must be simple and output must be displayed in an easy-to-understand format for better comprehension. For the web scraper tool, the search result displaying the product from multiple online stores will be properly presented and arranged such that the product price, product name, shop name and other relevant details can be seen and easily interpreted by users.

## 3.4.2 Logical Design

Logical design is a conceptual representation of the processes of the system. Logical design utilizes data flow diagrams, E-R diagrams, class diagrams, use-case diagrams, activity diagrams and sequence diagrams to represent the inputs, output, and flow of data.

### a) Class Diagram

Class diagram is an illustration of the classes and objects present in a system and the relationship existing between them. Below is a figure showing the class relationships of the web scraper tool. Class based diagrams are widely useful in applications running on Object oriented programming concepts. The class-based diagram shows the relationship between the User model, Product Model and Shop model.

-id: int

-name: string

-productUrl: int

-imageUrl: string

-price: int



Product

-id: int

-email: string

-Phone: int

-fullname: string



User

-changePassword()

-login()

-logout()

-id: int

-owner: User

-item: Product



SavedProduct

-item: Product

-category: string



Hotdeals

-addProduct()

-removeProduct()

Figure 3.1: Class diagram representing a web scraper tool for e-commerce

**b) Use Case Diagram**

Use case diagram illustrates the functionality, features, and requirements of the system and how users interact with the system to perform activities by using use cases and actors (Sabharwal, Kaur, & Sibal, 2017).

Figure 3.2: Use-case diagram for web scraper tool for e-commerce

Table 3.1 Use case narrative for user saving discovered optimal product after comparison

|  |  |
| --- | --- |
| Use Case 1 | User |
| Goal in content | User should be able to save optimal products after completing price comparison |
| Priority | High |
| Preconditions | User must be logged in |
| Post-condition (success end) | Save was successful |
| Post-condition (failure end) | Save was unsuccessful |
| Actor | User |
| Trigger | A user request to save optimal product |
| Description (event flow) | |  |  | | --- | --- | | Actor action | System response | | 1. Click the save Icon | 1. Display the request loading or processing animation and save the product | |  | 1. Display outcome of save either a success or failed | |

### c) Activity Diagram

Activity diagram is a graphical representation of the workflow of a system from one activity to another. These activities represent the workings or operations of the system. Activity diagrams are very important as they can be used to model the workflow and processes of a system. A properly designed activity diagram should be able to effectively show and communicate the dynamic behavior of a facet of the system.

Activity diagram uses a combination of shapes to show the flow of activities. The common shapes used are:

1. Black circle to represent the initial state (start of the workflow).
2. Encircled black circle to represent the final state (end of the workflow).
3. Rounded rectangles to represent actions.
4. Diamonds to represent decisions.

NO

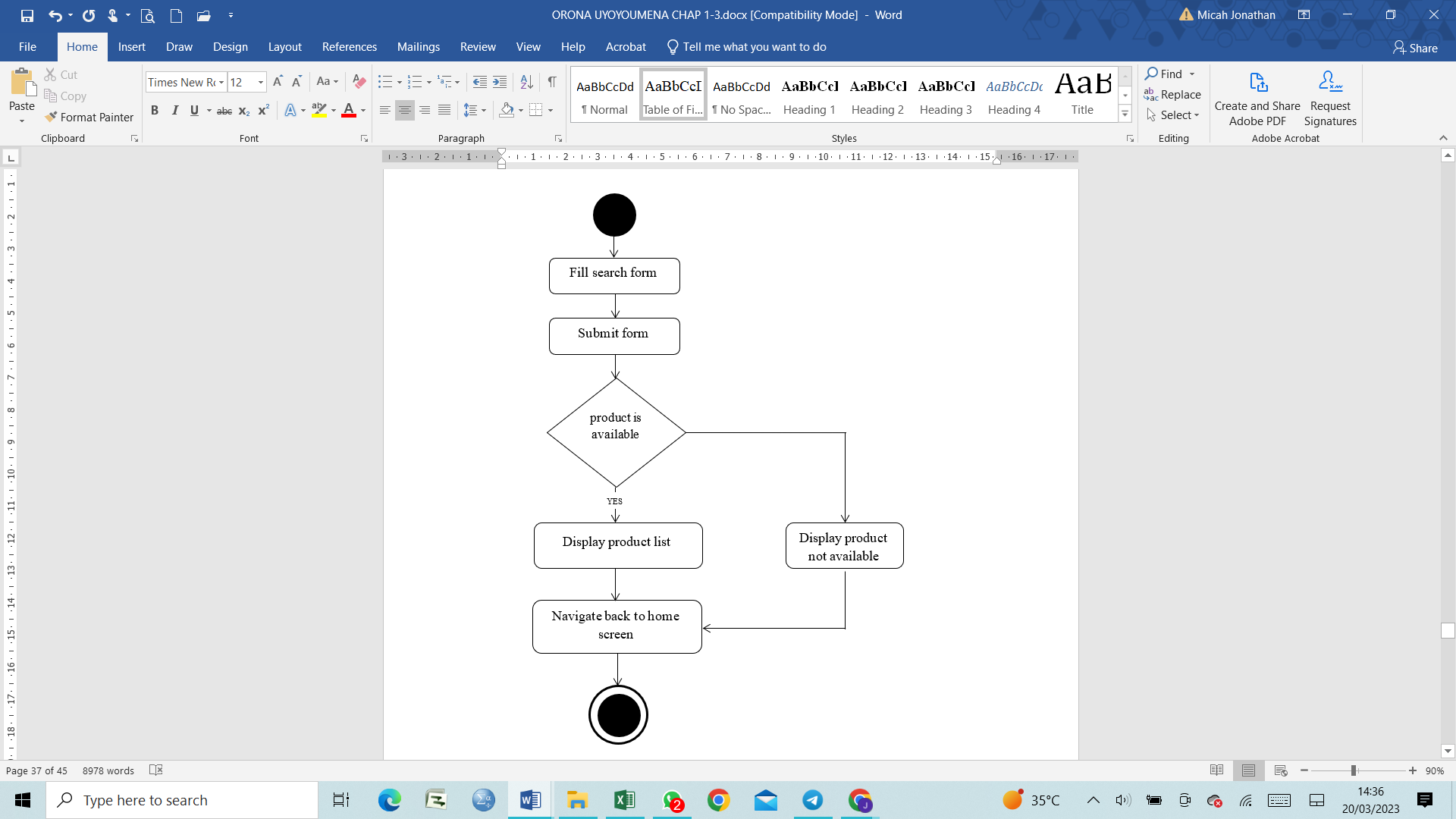


Figure 3.3: Activity diagram for a user searching a project

NO

Display product list

User clicks save button

User logged in

Redirect to login screen

Refresh list view

Display saved successfully

YES

Login

Figure 3.4: Activity diagram for user saving a product

### d) Sequence Diagram

Sequence diagram is an interactive diagram that is used to show the interactive behaviour of a system. They show how the system operations are conducted sequentially. Sequence diagrams focus on time to effectively illustrate the order of interactions in a system. Sequence diagrams are versatile because they show interactions that involve use cases, interactions between users and the system, interactions between subsystems, and even interactions between the system and other systems.

User

Web application

Web scraper

Search task (product name)

Scrape data task

Scraped result

alt

**Product found**

Display product list

**Product not found**

Display product not found

Figure 3.5: Sequence diagram for a user searching for a product

User

Web application

Sqlite Database

Save task (product name, id)

Save Product

Save Result

alt

**Save successful**

Display successful

**Save Unsuccessful**

Display not successful

Figure 3.6: Sequence diagram for user saving a discovered product

## 3.5 CONCEPTUAL DESIGN

Conceptual design ensures data security and integrity of the system as a whole. Database schema containing requirements for the database to avoid common database errors and loopholes in the database.

The web scraper tool for e-commerce stores uses the SQLite database to store data. SQLite is the default database for the Django web framework. SQLite is widely adopted because of its simplicity and robust nature. Django supports SQLite using object-relational mappers to perform CRUD queries to the database. This means Django code will be used to query the database instead of raw SQL statements.

### 3.5.1 Description of Tables

The web scraper tool for e-commerce stores comprises several database models and tables for holding data. Django models stored in a file called models.py contains model instances that each represent a Table in the SQLite database. Some of these Tables, their conditions, and their constraints are listed below:

1. **User Table**

The User table is the table that stores all the user data. The table below shows the fields, constraints, and conditions of the User Table.

Table 3.2 User table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Description** | **Data** **Type** | **Null** | **Key** |
| \_id | Unique user identifier | Integer | No | PRIMARY |
| Full Name | Full name of the user | Text | No |  |
| Location | Location of the user | Text | Yes |  |
| Data registered | Date signed up to the web application | Text | No |  |
| Email | Email of the user | Text | Yes |  |
| Phone | Phone Number of the user | Integer | Yes |  |

**ii. Product Table**

The Product table is the table that stores all the product data. The table below shows the fields, constraints, and conditions of the Product Table.

Table 3.3 Product table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Description** | **Data** **Type** | **Null** | **Key** |
| \_id | Unique product identifier | Integer | No | PRIMARY |
| Product Name | Name of the product | Text | No |  |
| Link | Merchant Link to Purchase product | Text | No |  |
| Store | Merchant store or owner of product | Text | No |  |
| Available | Field to tell if the product is still available | Boolean | No |  |

# REFERENCES

Anjali, G. (2014). E-commerce: Role of E-commerce in today's. *International Journal of Computing and Corporate Research*, 1.

Blanckenberg, N. (2022, August 31). *Competitive pricing strategies for ecommerce*. Retrieved 2022, from business2community: https://www.business2community.com/ecommerce/competitive-pricing-strategies-for-ecommerce-02286778

Bonastre, L. (2014). A set of heuristics for user experience evaluation in e-commerce websites. *The Seventh International Conference on Advances in Computer-Human Interactions*, 27-28.

Brown, I., & Jayakody, R. (2008). B2C e-Commerce Success: a Test and Validation of a Revised Conceptual Model . *Electronic Journal Information Systems Evaluation,Volume 11,Issue 3* , 167.

Cem, D. (2022, October 26). *Web scraping in e-commerce: Use cases and challenges*. Retrieved November 18, 2022, from research AI multiple: https://research.aimultiple.com/web-scraping-ecommerce

Comparison of Web Scraping Techniques:Regular Expression, HTML DOM and Xpath . (2018). *International Conference on Industrial Enterprise and System Engineering*, 283.

Coppola, D. (2022, July 27). *Number of Amazon.com employees from 2007 to 2021*. Retrieved from Statista.com: https://www.statista.com/statistics/234488/number-of-amazon-employees/

Doan, R. J., & Simon, H. (1996). *Power Pricing.* Free Press Publishings.

Geri, M. (2022, June 24). *A Comprehensive Guide to the World’s Top 50 Ecommerce Companies*. Retrieved November 2022, from influentialmarketinghub.com: https://influencermarketinghub.com/top-ecommerce-companies/

Gorton, I. (2011). *Essential Software Architecture.* Springer Publishers.

Kasereka, H. (2021). Importance of web scraping in e-commerce. 1-3.

Khder, A. M. (2021). Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application. *International Journal of Advances in Soft Computing & Applications*, 147.

Lanford, P. (2006). E-Commerce: A Trust Perspective. *International Conference on Internet Computing*, 1-3.

Majest, E. (2021, November 17). *Why is data important for your business?* Retrieved 11 30, 2022, from majesteye.com: https://www.majesteye.com/why-is-data-important-for-your-business/

Ozsahan, H. (2022). *12 Best Web Scraping Tools in 2022 to Extract Online Data*. Retrieved 11 30, 2022, from popupsmart: https://popupsmart.com/blog/web-scraping-tools

Raul, H. (2014, October 22). *Top 7 Techniques of Web Scraping Only Experts Use*. Retrieved December 1, 2022, from loginworks: https://www.loginworks.com/blogs/160-techniques-of-web-scraping

Rohmat, G., & Alam, R. (2018). Comparison of Web Scraping Techniques: Regular Expression, HTML DOM and Xpath. *International Conference on Industrial Enterprise and System Engineering* , 283.

Ron, A. (2021, December 4). *Google is killing the Google Shopping app*. Retrieved November 2022, from Arstechnica: https://arstechnica.com/gadgets/2021/04/google-is-killing-the-google-shopping-app/

Sabharwal, S., Kaur, P., & Sibal, R. (2017). Empirical and Theoretical Validation of a Use Case Diagram Complexity Metric. *I.J. Information Technology and Computer Science*, 36.

Salinesi, C., & Regnell, B. (2002). Proceedings of the 8th International Working Conference on Requirements Engineering. *Foundation for Software Quality (REFSQ'02)*, 102-107.

Shalini, A. (2022). E-Commerce Analysis and Product Price Comparison Using Web. *International Journal of Research Publication and Reviews, Vol 3, no 6*, 2.

Shin, N. (2001). Strategies for competitive advantage in electronic commerce. *Journal of Electronic Commerce Research, VOL. 2*, 165.

Silviu, V. M. (2010). The premises and evolution of electronic commerce. *Academia*.

Sirisuriya, D. (2015). A Comparative Study on Web Scraping. *International Research Conference, KDU*, 137-138.

Vikram, K. (2020, September 22). *21st Century Workplace Automation : Beyond Bots!* Retrieved November 2022, from wire19: https://wire19.com/21st-century-workplace-automation-beyond-bots/

Zhao, B. (2017). Web scraping. *Encyclopedia of big data*, 1-2.