WEB SCRAPER FOR AMAZON E-COMMERCE WEBSITE

A MINI PROJECT REPORT

Submitted by

J. ABDUL RAHMAN (17CSR001)

G. KAVIN (17CSR046)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY,

(AUTONOMOUS)

ERODE - 638012

DECEMBER 2020

BONAFIDE CERTIFICATE

Certified that this project report **"WEB SCRAPER FOR AMAZON E-COMMERCE WEBSITE"** is the bonafide work of "**ABDUL RAHMAN J (17CSR001), KAVIN G (17CSR046)"** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Ms.S.Ponni M.E.,

SUPERVISOR

Assistant Professor,

Department of CSE,

Velalar College of Engineering and Technology,

Erode-638012.

SIGNATURE

Dr.S. Jabeen Begum M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Professor & Head,

Department of CSE,

Velalar College of Engineering and Technology,

Erode-638012.

Submitted for Semester Mini-Project viva-voce examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

On the glorious occasion of having consummated our mini project we would like to thank our honorable Secretary and Correspondent, Vellalar Educational Trust Thiru.S**.D.CHANDRASEKAR B.A**., for having provided ample facilities to complete the project successfully.

We express our deep sense of gratitude to our beloved Principal **Dr.M. JAYARAMAN Ph.D.,** Velalar College of Engineering and Technology for his patronage and encouragement.

We are endlessly indebted to our honorable Dean (Academic & Student Affairs) Prof.P. **JAYACHANDAR M.E.,** for giving us the opportunity and continuous inspiration to carry out this project.

We solemnly express our heartiest gratitude to our Head of the Department **Dr S. JABEEN BEGUM M.E., Ph.D.,** for her valuable guidance and encouragement.

We thank our Project Coordinator **Dr.S. KAYALVILI M.E., M.B.A., Ph.D.,** Associate Professor and our Project Supervisor **S. PONNI M.E.,** Assistant Professor for their technical support, guidance and encouragement in all aspects throughout the work.

We profoundly thank all teaching and non-teaching staff members of Computer Science and Engineering department. We also extend our thanks to our beloved parents and friends.

ABSTRACT

Web scraper extracts data available on the Amazon e-commerce website in a format that makes it easier to collect and use it, for example in the form of downloadable CSV. For instance, e-commerce web scrapers help in the identification of consumer preferences and choices. They help in assessing trends of purchase behavior in online circles. Web scraping aided several e-commerce companies since years including Amazon, Walmart, Shopify, eBay and many more online stores.

BeautifulSoup is a parsing library which also does a pretty good job of fetching contents from URL and allows to parse certain parts of them without any hassle. It only fetches the contents of the URL and then stops. It does not crawl unless manually put it inside an infinite loop with certain criteria.

Scrapy is a fast, open-source web crawling framework written in Python, used to extract the data from the web page with the help of selectors based on XPath. Scrapy is easily extensible, fast, and powerful. Scrapy requests are scheduled and processed asynchronously. It is possible to scrap any website, though that website does not have API for raw data access.

Python Scrapy spider that searches Amazon for a particular keyword, extracts each products ASIN ID and scrape all the main information from the product page. The spider will iterate through all pages returned by the keyword query.

LIST OF ABREVIATIONS

CSV	-	Comma Seperated Values
URL	-	Uniform Resource Locator
API	-	Application Programming Interface
XPATH	-	XML Path Language

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 WEB SCRAPING

The term “web scraping” has several synonyms. The two other popular names are “web harvesting” and “web data extraction”. All three aliases are quite self-explanatory showing that it have to deal with extracting data from all possible websites. Web scraping software may access the World Wide Web directly using the Hypertext Transfer Protocol, or through a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web crawler. Web scraping a web page involves fetching it and extracting from it.

The content of a page may be parsed, searched, reformatted, its data copied into a spreadsheet, and so on. Web scrapers typically take something out of a page, to make use of it for another purpose somewhere else. An example would be to find and copy names and telephone numbers, or companies and their URLs, to a list (contact scraping). Is this legal? Of course, yes, because web scraping is used to collect only visible information that is displayed to store visitors. No private data is stolen, so there are no reasons to be afraid for reputation or any law violations.

1.1.1 THE CRAWLER

A web crawler, which is generally called as “spider,” is an artificial intelligence that browses the internet to index and search for content by following links and exploring, like a person with too much time on their hands. In many projects first “crawl” the web or one specific website to discover URLs which then pass on to the scraper.

1.1.2 THE SCRAPER

A web scraper is a specialized tool designed to accurately and quickly extract data from a web page. Web scrapers vary widely in design and complexity, depending on the project. An important part of every scraper is the data locators (or selectors) that are used to find the data that want to extract from the HTML file - usually xpath, css selectors, regex or a combination of them is applied.

1.2 WEB SCRAPING E-COMMERCE WEBSITE

E-commerce web scraping is a kind of data scraping related to the e-commerce segment of the web. If it is needed to get all possible information about goods and services available on a competitor's website, use e-commerce web scraping. Copying and pasting all data into a spreadsheet manually, but this process is attended by the risk of wasting tons of time and effort, especially when there are several competitors with a plethora of goods.

The global e-commerce market in the present day consists of more than millions of stores and that number is growing every day with the emergence of other businesses. In this case, manual tracking to assess and optimize product prices, which are neither practical nor feasible. But is it possible to extract ecommerce web data in real time? Absolutely!! Use dynamic web scraping to build Competitor price tracking software or a chrome extension that minimizes time spent on monitoring competitor prices by providing a dynamic pricing strategy in real time.

1.2.1 WEB SCRAPING PROCESS

1. Identifying target website.
2. Collect URLs of the pages to extract data from.
3. Make request to these URLs to get the HTML of the page.
4. Use locators to find the data in the HTML.
5. Save the data in a JSON or CSV file or some other structured format.

1.2.2 PREDICTIVE ANALYSIS

Predictive analysis helps E-Commerce merchants to know what customers want and how much they will pay for it, formulate targeted recommendations and promotions, improve supply chain management and monetize and make more profit. With the help of data scraping, millions of data obtain with exposure and then use predictive analysis to analyze past and present trends to foretell the future trend and leverage that huge competitive advantage to maximize sales.

1.2.3 PRICE-MONITORING AND PRODUCT RESEARCH

Price Monitoring is one of the most common advantages of web scraping for ecommerce websites. Each company gains from price monitoring of its rival websites ranging from eBay sellers to Amazon retailers, who use web scraping for the same. Companies may figure out prices for the same good or services in various places. This offers the client a chance to settle about its own offering's cost.

1.3 SCRAPING AMAZON WEBSITE

Amazon has been on the cutting edge of collecting, storing, and analyzing a large amount of data. Be it customer data, product information, data about retailers, or even information on the general market trends. Since Amazon is one of the largest e-commerce websites, a lot of analysts and firms depend on the data extracted from here to derive actionable insights.

This data is called alternative data and can be derived from multiple sources. Some of the most prominent sources of alternative data in the e-commerce industry are customer reviews, product information, and even geographical data. E-commerce websites are a great source for a lot of these data elements.

1.3.1 DATA SCRAPING FROM AMAZON WEBSITE

a) ASIN ID
b) Product name
c) Product rating
d) Number of reviews
e) Product price

1.4 OBJECTIVE

The main objective of the system is to scrape data from e-commerce website which helps companies to find out charges from different sites for the same product or service and this gives the business an opportunity to decide their own offering's price. Price optimization helps e-commerce businesses to boost profits tremendously.

Many websites have large collections of pages generated dynamically from an underlying structured source like a database. Data of the same category are typically encoded into similar pages by a common script or template. In data mining, a program that detects such templates in a particular information source, extracts its content and translates it into a relational form, is called a wrapper. Wrapper generation algorithms assume that input pages of a wrapper induction system conform to a common template and that they can be easily identified in terms of a URL common scheme. Moreover, some semi-structured data query languages, such as XQuery and the HTQL, can be used to parse HTML pages and to retrieve and transform page content.

The cleanest and most popular way to retrieve Amazon product pages is to use their ASIN ID.ASIN’s are a unique ID that every product on Amazon has. The ASIN value can be extracted from the product listing page by using Scrapy’s built-in XPath selector extractor methods.

CHAPTER 2

LITERATURE REVIEW

Web scraping, the extraction of patterned data from web pages on the internet, has been developed in the private sector for business purposes and it offers substantial benefits. By building and sharing protocols that extract search results and other data from web pages, those looking for grey literature can drastically increase their transparency and resource efficiency. One major current use for web scraping is for businesses to track pricing activities of their competitors: pricing can be established across an entire site in relatively short time scales and with minimal manual effort. Various other commercial drivers have caused a large number and variety of web scraping programs to have been developed in recent years. Web scrapers are an attractive technological development. The availability of a wide range of free and low-cost web scraping software provides an opportunity for significant benefits to those with limited resources, particularly researchers working alone or small organization.

2.1 Non-Functional Requirement

Considering the current market situation of e-commerce in Europe with respect to the number of e-shoppers and online retailers as well as the diversity of popular product categories the following non-functional requirements can be derived:

1) The estimation of 645,000 e-shop websites and 264 million European e-shoppers there is intensive competition within Europe. The internet prices are updated daily or even more frequent. Since there are many competitive e-shops and products to monitor almost in a daily frequency the approach needs to be fully automated.

2) More than 50% of the e-shoppers from United Kingdom (54%) and the Nordic Region (56%) are buying their goods from foreign e-shop websites and nearly the half of the e-shoppers from Germany (46%), France (43%) and Spain (43%) are purchasing products from external e-shops. Thus, there is a cross-border competition within European e-commerce what leads to the need of a language independent approach.

3) The survey also shows that the domains of the top-selling products sold online are very diverse. This leads to the necessity for an approach which is independent from a specific product domain.

2.2 Functional Requirements

In order to be able to automatically identify, extract and store the attributes of all products of any e-shop website in a structured format the approach needs to provide the following functions in a sequential order:

1) **Product page detection:** Automated detection of all Web pages containing product lists within the e-shop websites.

2) **Product record detection:** Automated detection of the product lists and the single product records within the Web pages.

3) **Product attribute extraction:** Automated detection, extraction and structured storing of the product attributes of the product records.

4) **Product attribute assignment:** Automated assignment of the extracted attributes to a defined product property for further processing steps and calculations.

RELATED WORK

There are several comparison shopping engines (CSEs) available on the Web for comparing products and prices of different e-shops. Some comparison shopping engines even offer access to their data through an application programming interface (API). However, recent comparison shopping engines obtain their data directly from the online retailers through a specific product feed defined by the comparison shopping engine operators. On this account there are only the products and prices of those e-shop websites available in the comparison shopping engines which e-shop owners have provided data to the engine. Thus, a price comparison based on a comparison shopping engine is limited to the data of e-shops available in the engine. More flexible solutions are able to collect the product and price data of arbitrary e-shop websites. Over the years several different approaches for identifying and extracting product records or even structured product attributes from the Web have been developed. These existing approaches range from scientific methodologies to complete ready-to-use software services.

2.3 Ready-to-use Software Tools and Services

Examples for ready-made software services to extract structured data from the Web are Kimono Labs1, import.io2 and Crawlbot3. Kimono Labs is a tool for the fast creation of APIs from websites. The tool is provided in the form of a Bookmarklet, which is a small software macro for a Web browser written in JavaScript, or as a browser plug-in for the Google Chrome browser. The Kimono Labs tool offers a graphical interface through a Web browser to mark the data which shall be extracted. It exploits the Cascading Style Sheets (CSS) selectors of the Hypertext Markup Language (HTML) elements and uses regular expressions for extracting the data marked by the user. The data is collected and provided as a JavaScript Object Notation (JSON) file or as a comma-separated values (CSV) sheet. The tool import.io is available in the form of a Web service or as a desktop application. Import.io needs the input of a Uniform Resource Locator (URL) to a list page, it analyses the page and returns its data as a table, CSV sheet or it offers a Web service API to the data. Crawlbot analyses the detail pages of products and returns a set of product attributes (e.g. product name, regular product price) available on the detail page in structured JSON format.

2.4 Scientific Approaches

Numerous scientific approaches for identifying and extracting data records or data attributes from the Web have been developed. The Mining Data Records in Web Pages (MDR) algorithm discovers data regions in Web pages by comparing the child nodes of each node within an HTML tree starting at the root node and identifying similar node combinations. The Visual information aNd Tag structure based wrapper generator (ViNTs) tool introduced automatically generates wrappers for extracting search result records of arbitrary search engines. ViNTs compares the content and the structure of some example result pages as well as an empty result page of a search engine for identifying and extracting its result records. An approach for extracting structured product specifications from producer websites. For the detection of the producers' websites a keyword search is performed by using a common search engine. The product data is extracted as keyvalue pairs by executing three different wrapper induction algorithms. The approach described is based on Visual Block Model (VBM) which is a product of the HTML tree and the CSS of a Web page and is built by the rendering process of a layout engine. The approach filters the basic blocks (visual blocks containing other visual blocks) and calculates their visual similarity as well as their content similarity. For identifying and extracting the product records a basic block in the middle of the Web page is selected as a seed and all similar blocks around the seed are considered and extracted as product records.

2.5 Weaknesses of the Existing Approaches

The main weaknesses of the existing approaches can be found in meeting the functional requirements which are necessary for the fully automated identification and extraction of product attributes from e-shop websites. The step of assigning the extracted product attributes to pre-defined product properties is essential for being able to process the extracted data in further analysis steps, e.g. for calculating adequate product prices fitting the market situation. Only the approach proposed is able to detect the Web pages within a website which contain the product description. However, the problem of the approach presented in the need for key phrases which have to be provided by a user for each product domain in order to be able to work. Another issue with this approach is that the key phrases given by the users need to fit the phrases of the product detail page in order to obtain good results. Only Crawlbot assigns the extracted product attributes to pre-defined product properties for being able to immediately run further process steps on the extracted data. All the other approaches which partially meet that requirement need manual steps from the user in previous steps for being able to assign the product attributes to pre-defined properties. Unfortunately, Crawlbot is only able to extract data directly from a product's detail page and it is not able to process Web pages listing more than one product.

CHAPTER 3

SYSTEM STUDY

3.1 EXISTING SYSTEM

Beautiful Soup is one of the most popular Python libraries which helps in parsing HTML or XML documents into a tree structure to find and extract data. This tool features a simple, Python interface and automatic encoding conversion to make it easy to work with website data. This library provides simple methods and Python idioms for navigating, searching, and modifying a parse tree, and automatically converts incoming documents to Unicode and outgoing documents to UTF-8. But the problem with Beautiful Soup is it can't able to do the entire job on its own. This library requires specific modules to work done. It is difficult to migrate from one project to another using Beautiful Soup. It only fetches the contents of the URL and then stops. It does not crawl unless manually put it inside an infinite loop with certain criteria.

DRAWBACKS

- To analyze the retrieved data, it needs to be treated first. This often becomes a time-consuming work.
- For those, who are not much tech-savvy and aren't an expert, web scraping can be a confusing process. Even though, it's not a major issue.
- Sometimes **web scraping services** take time to become familiar with the core application and need to adjust to the scrapping language.
- Most web scrapping services are slower than API calls and another problem is the websites that do not allow screen scrapping.

3.2 PROPOSED SYSTEM

The core concept of the scraper development with Scrapy is the “Web Spider” called scrapers. These are small programs based on Scrapy. Each **spider is programmed to scrape a specific website** and shimmy from side to side like the eponymous spider. Object-oriented programming is used here: Each spider is its own Python class.

The architecture of the tool **is based on the needs of professional projects**. Scrapy contains an integrated pipeline for processing the scraped data. The page fetch in Scrapy is asynchronous; this means that several pages can be downloaded in parallel. Thus, Scrapy is well suited for scraping projects with a high volume of pages to be processed.

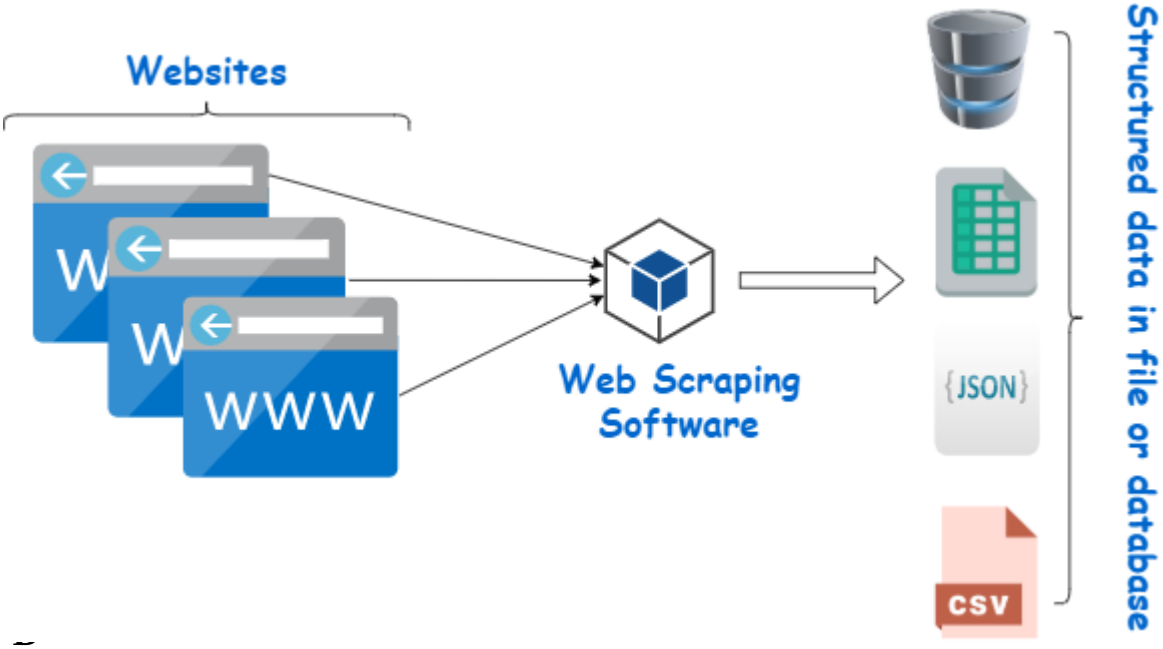


Figure 3.1 DIAGRAM OF PROPOSED SYSTEM

Advantages

- Web scraping helps in deriving accurate details.
- This gives the business an opportunity to decide their own offering’s price.
- Price-monitoring and Product Research.
- Better Customer analysis.
- Influences Marketing and Sales Strategy.
- Helps in Future analysis.

CHAPTER 4

PROPOSED METHODOLOGY

4.1 SCRAPY

Scrapy is a fast high-level web crawling and web scraping framework, used to crawl websites and extract structured data from their pages. It can be used for a wide range of purposes, from data mining to monitoring and automated testing.

Scrapy project architecture is built around “spiders”, which are self-contained crawlers that are given a set of instructions. It is easier to build and scale large crawling projects by allowing developers to reuse their code. Scrapy also provides a web-crawling shell, which can be used by developers to test their assumptions on a site’s behavior.

4.1.1 SPIDERS

Spiders are classes which define how a certain site (or a group of sites) will be scraped, including how to perform the crawl (i.e. follow links) and how to extract structured data from their pages (i.e. scraping items).

4.1.2 SELECTOR

Scrapy Selectors is a thin wrapper around parsel library, the purpose of this wrapper is to provide better integration with Scrapy Response objects. Scrapy comes with its own mechanism for extracting data. They are called selectors because they “select” certain parts of the HTML document specified either by XPath or CSS expressions.

4.1.3 ITEM PIPELINE

After an item has been scraped by a spider, it is sent to the Item Pipeline which processes it through several components that are executed sequentially.

Each item pipeline component (sometimes referred as just “Item Pipeline”) is a Python class that implements a simple method. They receive an item and perform an action over it, also deciding if the item should continue through the pipeline or be dropped and no longer processed.

Typical uses of item pipelines are:

- cleansing HTML data
- validating scraped data (checking that the items contain certain fields)
- checking for duplicates (and dropping them)
- storing the scraped item in a database

4.2 HTML

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as `<img />` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

4.3 JSON

JSON (JavaScript Object Notation) is an open standard file format, and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format, with a diverse range of applications, such as serving as a replacement for XML in AJAX systems.

JSON is a language-independent data format. It was derived from JavaScript, but many modern programming languages include code to generate and parse JSON-format data. The official Internet media type for JSON is application/json. JSON filenames use the extension .json.

4.4 XML

XML stands for Extensible Markup Language. XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

4.5 XPATH

It defines a language to find information in an XML file. It is used to traverse elements and attributes of an XML document. XPath provides various types of expressions which can be used to enquire relevant information from the XML document.

Xpath specification specifies seven types of nodes which can be the output of execution of the Xpath expression.

- Root
- Element
- Text
- Attribute
- Comment
- Processing Instruction
- Namespace

4.6 CSV FILE

CSV stands for "comma-separated values". is a delimited text file that uses a comma to separate values. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. A CSV file typically stores tabular data (numbers and text) in plain text, in which case each line will have the same number of fields.

CHAPTER 5

SYSTEM SPECIFICATION

5.1 HARDWARE REQUIREMENTS

This section gives the details and specification of the hardware on which the system is expected to work.

Processor	:	Intel dual core processor
RAM	:	2 GB SD RAM
Hard disk	:	500 GB

5.2 SOFTWARE REQUIREMENTS

This section gives the details of the software that are used for the development.

Operating System	:	Windows 7 and above or Linux
Environment	:	Python Editor
Language	:	Python-Language

5.2.1 PYTHON

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

CHAPTER 6

SYSTEM TESTING

Testing is a process of checking whether the developed system is working according to the original objectives and requirements. Testing is a set of activities that can be planned in advance and conducted systematically. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the global will be successfully achieved. In adequate testing if not testing leads to errors that may not appear even many months. This creates two problems,

- The time lag between the cause and the appearance of the problem.
- The effect of the system errors on the files and records within the system.
- A small system error can conceivably explode into a much larger Problem.

Effective testing early in the purpose translates directly into long term cost savings from a reduced number of errors. Another reason for system testing is its utility, as a user-oriented vehicle before implementation. The best programs are worthless if it produces the correct outputs. No other test can be more crucial. Following this step, a variety of tests are conducted.

- Unit testing
- Integration testing
- Validation testing

6.1 UNIT TESTING

Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class or derived/child class. Unit Testing is normally performed by software developers themselves or their peers. Unit testing increases confidence in changing/maintaining code.

Table 6.1 Levels of Testing

Levels	Testing
Level 0	Unit Testing
Level 1	Integration Testing
Level 2	System Testing
Level 3	Acceptance Testing

If good unit tests are written and if they are run every time any code is changed, the likelihood of any defects due to the change being promptly caught is very high. Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse.

Table 6.2 Test cases for Unit Testing

Test case no	Description	Excepted result
1	Test for all modules	All modules should communicate in the group.
2	Test for various functions in the framework	The result after execution should give the accurate result.

FUNCTIONAL TESTING

In functional testing need check the each components are functioning as expected or not, so it is also called as "**Component Testing**". Functional testing is to testing the functionality of the software application under test. Basically, it is to check the basic functionality mentioned in the functional specification document. Also check whether software application is meeting the user expectations. Also say that checking the behavior of the software application against test specification. What is all need to be check in Functional Testing?

- Is software is functioning as it should do?
- Is software is not functioning as it should not do?
- Is software is not doing as it not intended to do?

6.2 INTEGRATION TESTING

Integration Testing is a level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.

Although each software module is unit tested, defects still exist for various reasons like

A) A Module, in general, is designed by an individual software developer whose understanding and programming logic may differ from other programmers. Integration Testing becomes necessary to verify the software modules work in unity

B) At the time of module development, there are wide chances of change in requirements by the clients. These new requirements may not be unit tested and hence system integration Testing becomes necessary.

C) Interfaces of the software modules with the database could be erroneous

D) External Hardware interfaces, if any, could be erroneous

E) Inadequate exception handling could cause issues.

6.3 VALIDATION TESTING

Verification and Validation are the activities performed to improve the quality and reliability of the system and assure the product satisfies the customer needs. Verification assures the product of each development phase meets their respective requirements. Validation assures the final product meets the client requirements.

6.3.1 VERIFICATION

It is the process of evaluating a system or component to determine whether the product of a given phase satisfies the conditions imposed at the start of that phase. It is a Low level activity performed during development on key artifacts, like walkthroughs, reviews and inspections, mentor

feedback, training, checklists and standards; it demonstrates consistency, completeness, and correctness of the software at each stage and between each stage of the development life cycle.

6.3.2 VALIDATION

It is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements It is a High level activity performed after a work product is produced against established criteria ensuring that the product integrates correctly into the environment, it determines correctness of the final software product by a development project with respect to the user needs and requirements.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

As the Internet has grown astronomically and businesses have become increasingly dependent on data, it is now a compulsion to have access to the latest data on every given subject. Data has become the basis of all decision-making processes whether it's a business or a non-profit organization. Therefore, web scraping has found its applications in every endeavor of note in contemporary times. It is also becoming increasingly clear that those who will make creative and advanced use of web scraping tool will race ahead of others and gain a competitive advantage.

It is safe to say that web scraping has become an essential skill to acquire in today's digital world, not only for tech companies and not only for technical positions. On one side, compiling large datasets are fundamental to Big Data analytics, Machine Learning, and Artificial Intelligence; on the other side, with the explosion of digital information, Big Data is becoming much easier to access than ever.

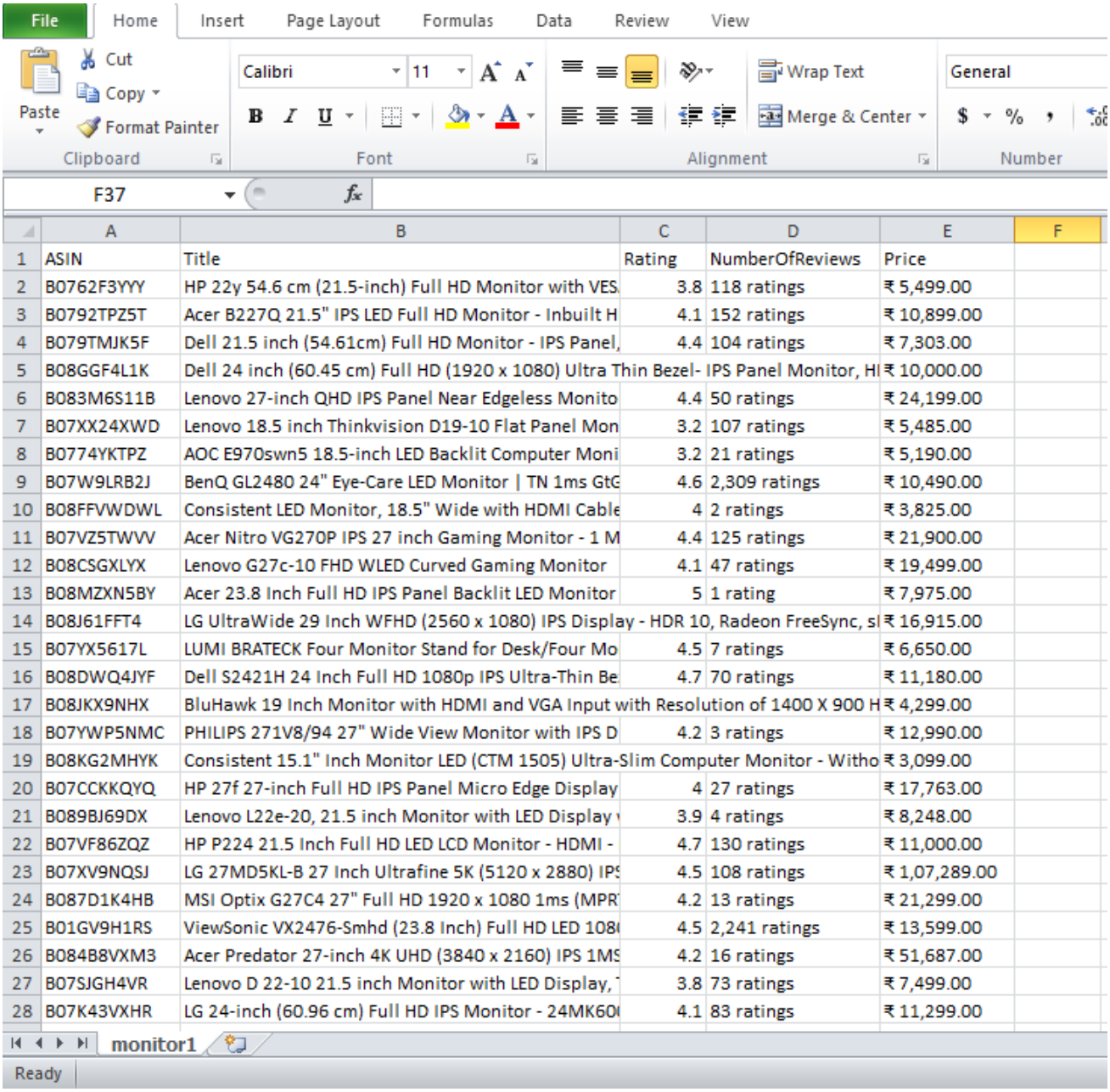
APPENDIX

SOURCE CODE :

```
#!/usr/bin/env python

import scrapy
from urllib.parse import urlencode
from urllib.parse import urljoin
import re
import json

queries = [ 'monitor' ]
API = 'key'

def get_url(url):
    payload = {'api_key': API, 'url': url, 'country_code': 'in'}
    proxy_url = 'http://api.scraperapi.com/?' + urlencode(payload)
    return proxy_url

class AmazonSpider(scrapy.Spider):
    name = 'amazon'
    def start_requests(self):
        for query in queries:
            url = 'https://www.amazon.in/s?' + urlencode({'k': query})
            yield scrapy.Request(url=get_url(url), callback=self.parse_keyword_response)

    def parse_keyword_response(self, response):
        products = response.xpath('//*[@data-asin]')
        for product in products:
            asin = product.xpath('@data-asin').extract_first()
            product_url = f"https://www.amazon.in/dp/{asin}"
            yield scrapy.Request(url=get_url(product_url), callback=self.parse_product_page,
meta={'asin': asin})
        next_page = response.xpath('//li[@class="a-last"]/a/@href').extract_first()
        if next_page:
            url = urljoin("https://www.amazon.in",next_page)
            yield scrapy.Request(url=get_url(url), callback=self.parse_keyword_response)
    def parse_product_page(self, response):
        asin = response.meta['asin']
        title = response.xpath('//*[@id="productTitle"]/text()').extract_first()
        rating = response.xpath('//*[@id="acrPopover"]/@title').extract_first()
        number_of_reviews =
response.xpath('//*[@id="acrCustomerReviewText"]/text()').extract_first()
        price = response.xpath('//*[@id="priceblock_ourprice"]/text()').extract_first()
        if not price:
            price = response.xpath('//*[@data-asin-price]/@data-asin-price').extract_first() or \
                    response.xpath('//*[@id="price_inside_buybox"]/text()').extract_first()
        yield {'asin': asin, 'Title': title,'Rating': rating, 'NumberOfReviews': number_of_reviews,'Price':
price}
```

SCREEN SHOT

The following picture is a screenshot of the excel sheet which has the data extracted from the Amazon e-commerce website .



File Home Insert Page Layout Formulas Data Review View

Cut Copy Format Painter Paste — Clipboard

Calibri 11 — Font

Wrap Text Merge & Center — Alignment

General — Number

F37

	A	B	C	D	E	F
1	ASIN	Title	Rating	NumberOfReviews	Price	
2	B0762F3YYY	HP 22y 54.6 cm (21.5-inch) Full HD Monitor with VES	3.8	118 ratings	₹ 5,499.00	
3	B0792TPZ5T	Acer B227Q 21.5" IPS LED Full HD Monitor - Inbuilt H	4.1	152 ratings	₹ 10,899.00	
4	B079TMJK5F	Dell 21.5 inch (54.61cm) Full HD Monitor - IPS Panel,	4.4	104 ratings	₹ 7,303.00	
5	B08GGF4L1K	Dell 24 inch (60.45 cm) Full HD (1920 x 1080) Ultra Thin Bezel- IPS Panel Monitor, HI			₹ 10,000.00	
6	B083M6S11B	Lenovo 27-inch QHD IPS Panel Near Edgeless Monito	4.4	50 ratings	₹ 24,199.00	
7	B07XX24XWD	Lenovo 18.5 inch Thinkvision D19-10 Flat Panel Mon	3.2	107 ratings	₹ 5,485.00	
8	B0774YKTPZ	AOC E970swn5 18.5-inch LED Backlit Computer Moni	3.2	21 ratings	₹ 5,190.00	
9	B07W9LRB2J	BenQ GL2480 24" Eye-Care LED Monitor | TN 1ms GtG	4.6	2,309 ratings	₹ 10,490.00	
10	B08FFVWDWL	Consistent LED Monitor, 18.5" Wide with HDMI Cable	4	2 ratings	₹ 3,825.00	
11	B07VZ5TWVV	Acer Nitro VG270P IPS 27 inch Gaming Monitor - 1 M	4.4	125 ratings	₹ 21,900.00	
12	B08CSGXLYX	Lenovo G27c-10 FHD WLED Curved Gaming Monitor	4.1	47 ratings	₹ 19,499.00	
13	B08MZXN5BY	Acer 23.8 Inch Full HD IPS Panel Backlit LED Monitor	5	1 rating	₹ 7,975.00	
14	B08J61FFT4	LG UltraWide 29 Inch WFHD (2560 x 1080) IPS Display - HDR 10, Radeon FreeSync, sl			₹ 16,915.00	
15	B07YX5617L	LUMI BRATECK Four Monitor Stand for Desk/Four Mo	4.5	7 ratings	₹ 6,650.00	
16	B08DWQ4JYF	Dell S2421H 24 Inch Full HD 1080p IPS Ultra-Thin Be	4.7	70 ratings	₹ 11,180.00	
17	B08JKX9NHX	BluHawk 19 Inch Monitor with HDMI and VGA Input with Resolution of 1400 X 900 H			₹ 4,299.00	
18	B07YWP5NMC	PHILIPS 271V8/94 27" Wide View Monitor with IPS D	4.2	3 ratings	₹ 12,990.00	
19	B08KG2MHYK	Consistent 15.1" Inch Monitor LED (CTM 1505) Ultra-Slim Computer Monitor - Witho			₹ 3,099.00	
20	B07CCKKQYQ	HP 27f 27-inch Full HD IPS Panel Micro Edge Display	4	27 ratings	₹ 17,763.00	
21	B089BJ69DX	Lenovo L22e-20, 21.5 inch Monitor with LED Display	3.9	4 ratings	₹ 8,248.00	
22	B07VF86ZQZ	HP P224 21.5 Inch Full HD LED LCD Monitor - HDMI -	4.7	130 ratings	₹ 11,000.00	
23	B07XV9NQSJ	LG 27MD5KL-B 27 Inch Ultrafine 5K (5120 x 2880) IPS	4.5	108 ratings	₹ 1,07,289.00	
24	B087D1K4HB	MSI Optix G27C4 27" Full HD 1920 x 1080 1ms (MPR	4.2	13 ratings	₹ 21,299.00	
25	B01GV9H1RS	ViewSonic VX2476-Smhd (23.8 Inch) Full HD LED 108	4.5	2,241 ratings	₹ 13,599.00	
26	B084B8VXM3	Acer Predator 27-inch 4K UHD (3840 x 2160) IPS 1MS	4.2	16 ratings	₹ 51,687.00	
27	B07SJGH4VR	Lenovo D 22-10 21.5 inch Monitor with LED Display,	3.8	73 ratings	₹ 7,499.00	
28	B07K43VXHR	LG 24-inch (60.96 cm) Full HD IPS Monitor - 24MK60	4.1	83 ratings	₹ 11,299.00	

monitor1

Ready

REFERENCES

[1] https://ieeexplore.ieee.org/document/7345488
[2] https://ieeexplore.ieee.org/document/7405846
[3] https://en.wikipedia.org/wiki/Web_scraping
[4] https://www.geeksforgeeks.org/project-idea-web-scraping/
[5] https://en.wikipedia.org/wiki/Data_scraping
[6] Deepak Kumar Mahto, Lisha Singh, A Dive into Web Scraper World, 2016 InternationalConference on Computing for SustainableGlobal Development (INDIACom), 978-9-3805-4421-2/16/$31.00 c , 2016 IEEE.
[7] List of Web Harvester, Data Scraper,Web Scraping Software and Tools, n.d.WebData Scraping. URL http://webdata-scraping.com/webscraping-software/
[8] S.C.M. de S Sirisuriya,2015, A Comparative Study on Web Scraping .Proceedings of 8th International Research Conference, KDU.