Restaurant Data Scraper: An Automated Tool for Extracting Restaurant Information Using Python Html, CSS and Selenium

Omkar Dhananjay Sagade
Dept. of computer Science engineering
MIT World Peace University
Pune, India
sagadeomkar11@gmail.com

Dhanuja Dhananjay Sagade
Dept. of computer Science engineering
MIT ADT University
Pune, India
Dhanujasagade31@gmail.com

Abstract*—In this paper, a web scraping solution for conveniently obtaining restaurant data from an HTML file—a Python-based Restaurant Data Scraper—is developed. BeautifulSoup is used for effective HTML parsing, and Selenium WebDriver is used to automate interactions with a web page. Python, HTML, CSS, Selenium, and BeautifulSoup are all used in the implementation. A structured CSV file containing the extracted data—such as restaurant names, ratings, descriptions, and addresses—allows for additional analysis. By offering a solution that can be modified for different restaurant listing formats, this project responds to the growing need for effective data extraction techniques from semi-structured HTML files. Scalable and UTF-8 encoded to handle special characters, the program provides a platform for more comprehensive data collection.

***Keywords*—Web scraping solution; Python-based Restaurant Data Scraper; CSV file ; HTML parsing; Data extraction techniques.**

I. Introduction

A custom web scraper created with Python is presented in this research project to effectively extract data from restaurant listing web pages. This is a crucial need considering the volume of unstructured data on digital platforms. With the help of BeautifulSoup for HTML parsing and Selenium for automated online navigation, the scraper efficiently collects structured data from both dynamic and static content, including restaurant names, reviews, ratings, locations, and pricing ranges. By allowing interaction with features such as pagination and "load more" buttons, Selenium manages dynamic content that isn't immediately apparent when a page loads. BeautifulSoup extracts
the pertinent data points by parsing the page content after Selenium has retrieved it. After parsing, the data is stored as a CSV file. producing an organised dataset that can be used for additional analysis in domains like as market research, consumer sentiment analysis, and cuisine trend tracking. By simplifying the process of converting intricate HTML into a format suitable for both quantitative and qualitative analysis, this Python-based method provides substantial benefits for insights based on data.

A. MOTIVATION:

The goal of our system project is to develop a system In order to meet the increasing demand for easily accessible, structured data from the enormous volume of unstructured restaurant information on the internet, the Restaurant Data Scraper was created. Rich data on restaurants can be found on digital platforms, but a large portion of this data is stored in HTML formats that make it difficult to use for direct analysis. Without the need for manual data entry, this project offers a simplified, automated method for extracting, processing, and organising this data, allowing academics, data analysts, and foodies to uncover insightful information. This program provides a high degree of accuracy and efficiency by automating the process using BeautifulSoup and Selenium, even when handling dynamically-loaded online material. Furthermore, The data's utility is expanded by its smooth export to CSV format, which makes it simple to share with colleagues and available for a variety of analytical tools. This scraper offers a workable way to handle web-based restaurant data effectively, turning it into a resource that may support sentiment analysis, market research, and industry trend predictions thanks to its simple setup and strong capabilities.

OBJECTIVES:

We will create a complete solution to streamline and automate the data extraction procedure from HTML files used for restaurant listings. To drastically cut down on the requirement for human data extraction, we will use Selenium to develop an automated application that navigates and interacts with the HTML files. BeautifulSoup will be used to precisely parse the text, identifying and extracting important information—including restaurant names, ratings, descriptions, and addresses—even from intricate HTML structures. The retrieved data will be saved in a structured CSV file, which will facilitate future research and make it simple to integrate with data analysis tools. UTF-8 encoding will be used to preserve data integrity, guaranteeing that special characters—like stars in ratings—are handled smoothly throughout the extraction and storage processes. Furthermore, we will create a reproducible workflow that includes detailed documentation, easy- to-follow installation instructions, and an example HTML website so that users can quickly duplicate the procedure.

Scope:

To meet the demands of researchers, market analysts, and data aficionados, the Restaurant Data Scraper project was created to simplify the extraction and organisation of restaurant data from HTML files. This application uses BeautifulSoup, Python, and Selenium to automatically gather restaurant data from dynamically-loaded webpages, including names, ratings, descriptions, and addresses. Conducting in- depth analyses such as competition research, consumer sentiment analysis, and market trend forecasts is made simple by exporting data in CSV format with UTF-8 encoding, which guarantees compatibility with a variety of analytical tools. The project's application area is expanded in domains that depend on structured data from unstructured web sources due to its high degree of adaptability and ability to be adjusted to scrape comparable data from other online platforms.

LITERATURE SURVEY

Many existing papers have delved into the field of web scraping through a myriad of methods. One such paper was made by Färholt, who conducted a controlled experiment to develop a web scraping technique using JavaScript's library, Puppeteer, while remaining undetectable to the websites being targeted. To facilitate this issue, web scraping has been introduced. Web scraping automates data extraction from web pages, which proves to be fast and effective. Data can be extracted using software services or self-built programs that run on websites with numerous levels of navigation. Web scraping is typically used by individuals and businesses that want to take advantage of the vast amount of freely available web data to make better decisions **[2]**

The usage of social networking sites and the internet is growing by the day, such as Facebook, Twitter, LinkedIn, and others; user knowledge is also growing on the internet, which is accessible from anywhere. This also gives hackers an edge when it comes to collecting information. From a commercial standpoint, social networking is critical in the development of the notion of growing revenue. It would aid customers in achieving rapid shopping and saving time, similar to internet shopping. Supporting the firm and earning from it, on the other hand, has advantages. Web scraping tactics utilised in the growth study will expose the researcher to a larger amount of data than is currently available in the data set, perhaps increasing the growth estimate. This topic was briefly discussed in the sections devoted to both of the examined items, but dealing with this point of view necessitates a concern about the current survey architecture, which does not require or only selectively permits the use of big data approaches within existing sampling frameworks**.[3]**

Web scraping is a process of information extraction from the world wide web (www), accomplished by writing automated script routines that request data by querying the desired web server and retrieving the data by using different parsing techniques. Scraping helps in transforming unstructured HTML data into various structured data formats like CSV, spreadsheets. As it is known, the nature of web data is changing frequently, using an easy-to-use language like python which accepts dynamic inputs can be highly productive, as code changes are easily done to keep up with the speed of web updates. **[1]**

Using the wide collection of python libraries, such as requests, pandas, csv, webdriver can ease the process of fetching URLs and pulling out information from web pages, building scrapers that can hop from one domain to another, gather information, and store that information for later use. To automate web browser interaction, the single interface open-source tool Selenium is used that can mimic human browsing behaviors**.**

The proposed experimental work shows, parsing the HTML code, installation of python and selenium, python scripting and interpretation, and structural extraction of web information. The evolving needs of internet and social media services require various techniques for the extraction of web data. Web information is mostly unstructured, the proposed work helps to organize the unstructured data and

make it useful for various data analysis techniques. There are plenty of techniques for web scrapping. **I.**

But selenium is all weather scrapping techniques. The problem of other scrapper techniques like Beautifulsoup, auto scrapper, html parsing etc. they are not feasible in all aspects.The first problem with beautifulsoup is if we encounter any HTTP error then we have to deal with it first and then we can go for scrapping.The most common problem with beautifulsoup and other scrapping techniques is that if a website has not allowed or permission of web crawling then it will not be going to scrap the data (only that data that has not allowed crawling).Then for scrapping this problem, Selenium is used.Selenium is quite popular in automation. It is used in JAVA and Python both. Selenium uses its own alias web browser.When we pass the link of any websites, it will open those websites in its own

web browser. Selenium pretend that website is opening on a web browser, so by this it can solve crawling problem which is happening in other web scrapping techniques. So, that's why it is all weather scrapping technique and it is invincible.**[4]**

PROJECT BACKGROUND

The increasing use of digital platforms for restaurant listings and reviews, where a wealth of important data is frequently kept in unstructured HTML formats, gave rise to the Restaurant Data Scraper project. Because manual data extraction is time- consuming and prone to mistakes, automated solutions are required. By automating the data extraction process and parsing complex HTML information into structured formats appropriate for analysis, this project makes use of Python, Selenium, and BeautifulSoup. The initiative bridges the gap between raw online data and actionable insights by offering a user-friendly, effective tool to extract features like restaurant names, ratings, descriptions, and addresses. Because of its versatility and scalability, it is an essential tool for market research, customer sentiment analysis, and culinary analytics.

IMPLEMENTATION

An HTML file (index.html) with restaurant listings is the first input used in the Restaurant Data Scraper project's implementation. This file is stored in the project directory so that the Python script (scraper.py) may easily access it. Starting with Selenium WebDriver, which opens the HTML file and engages with it, the script is made to automate the data extraction process. When managing dynamic material or carrying out operations like loading hidden items, Selenium is especially helpful in making sure that all pertinent information is available. BeautifulSoup takes over to analyze the HTML structure and extract particular data points, such restaurant names, ratings, descriptions, and addresses, as soon as the HTML file has finished loading. BeautifulSoup guarantee accurate data

extraction by focusing on the right HTML tags and characteristics. while reducing noise from extraneous components. To ensure that the output is correct and comprehensive, the script uses data processing and error handling techniques to clean the collected data, handle special characters (such as stars in ratings), and fix any missing or conflicting information. Ultimately, a CSV file called restaurants.csv contains the cleaned and organized data. This file has a structured format that makes it simple to use for additional analysis, visualization, or sharing with other people. In addition to providing flexibility for adaption to comparable web scraping jobs, the overall approach guarantees a smooth workflow by converting unstructured HTML data into actionable insights.

I. Proposed Methodology.

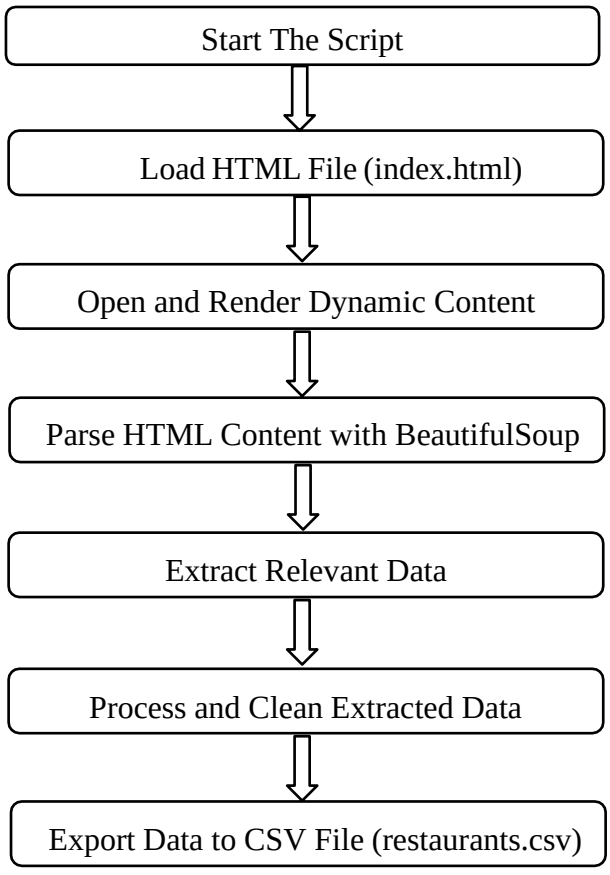


Fig. Execution of Modules

Architecture

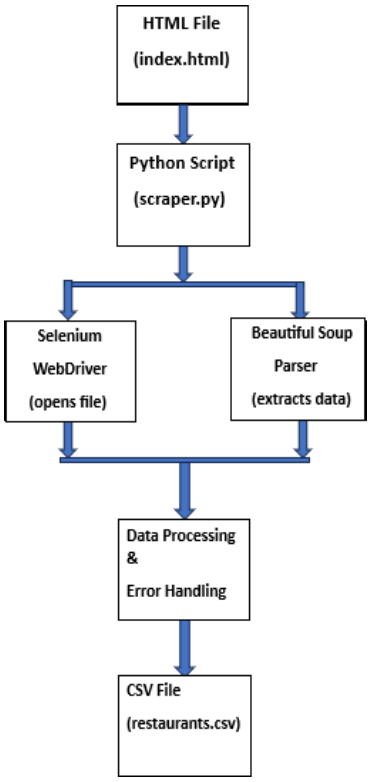
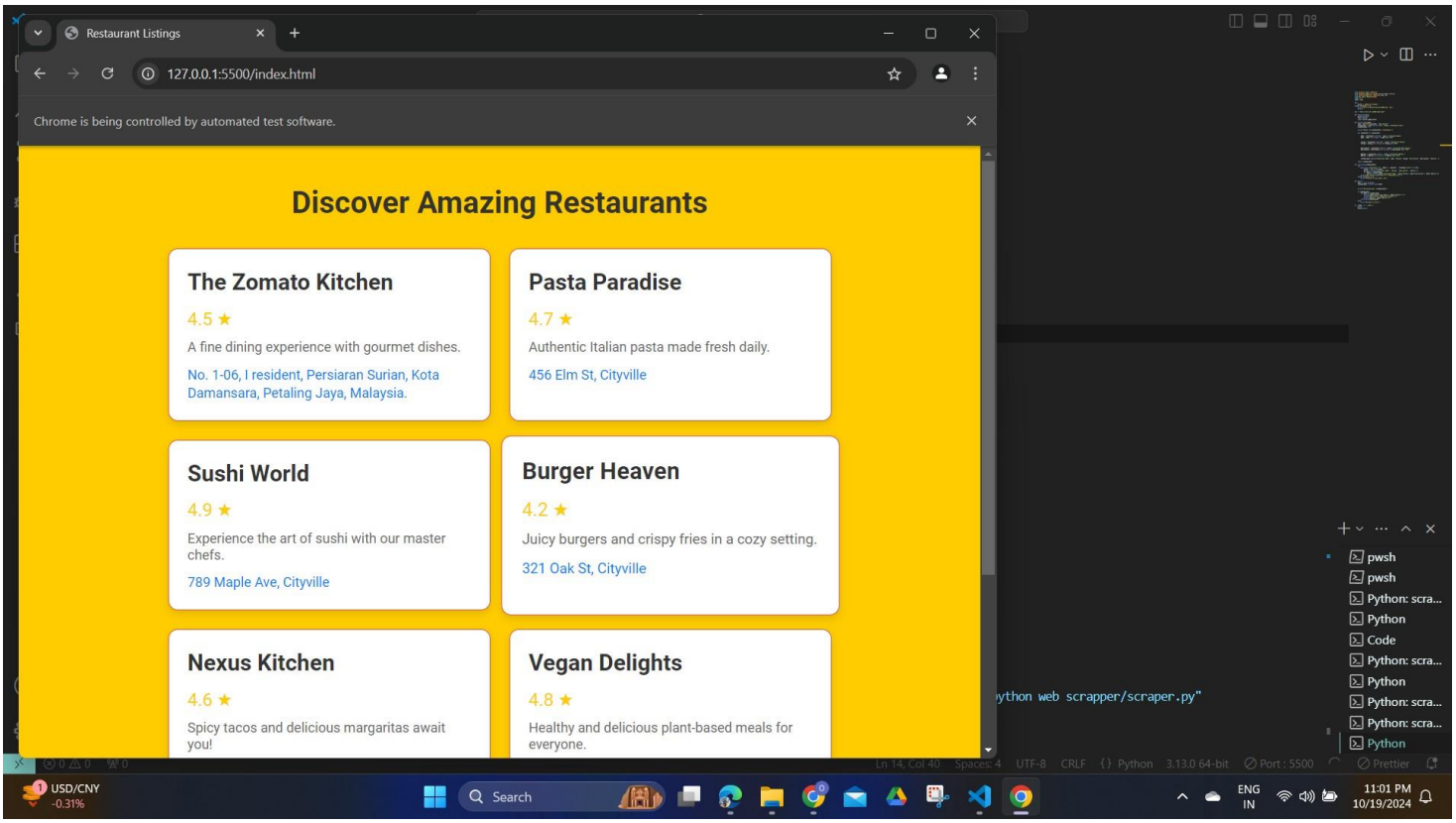


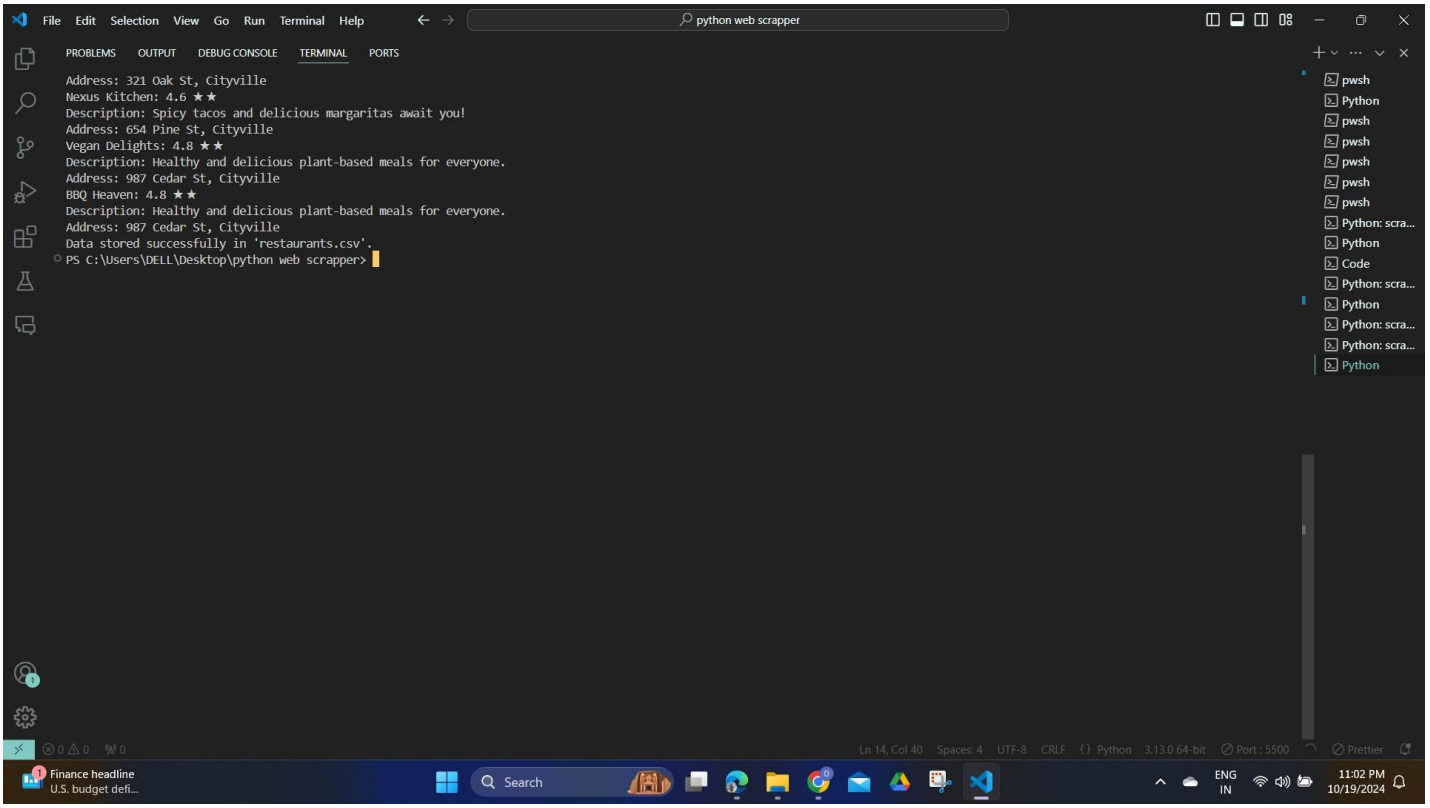
Fig. Architecture of Project

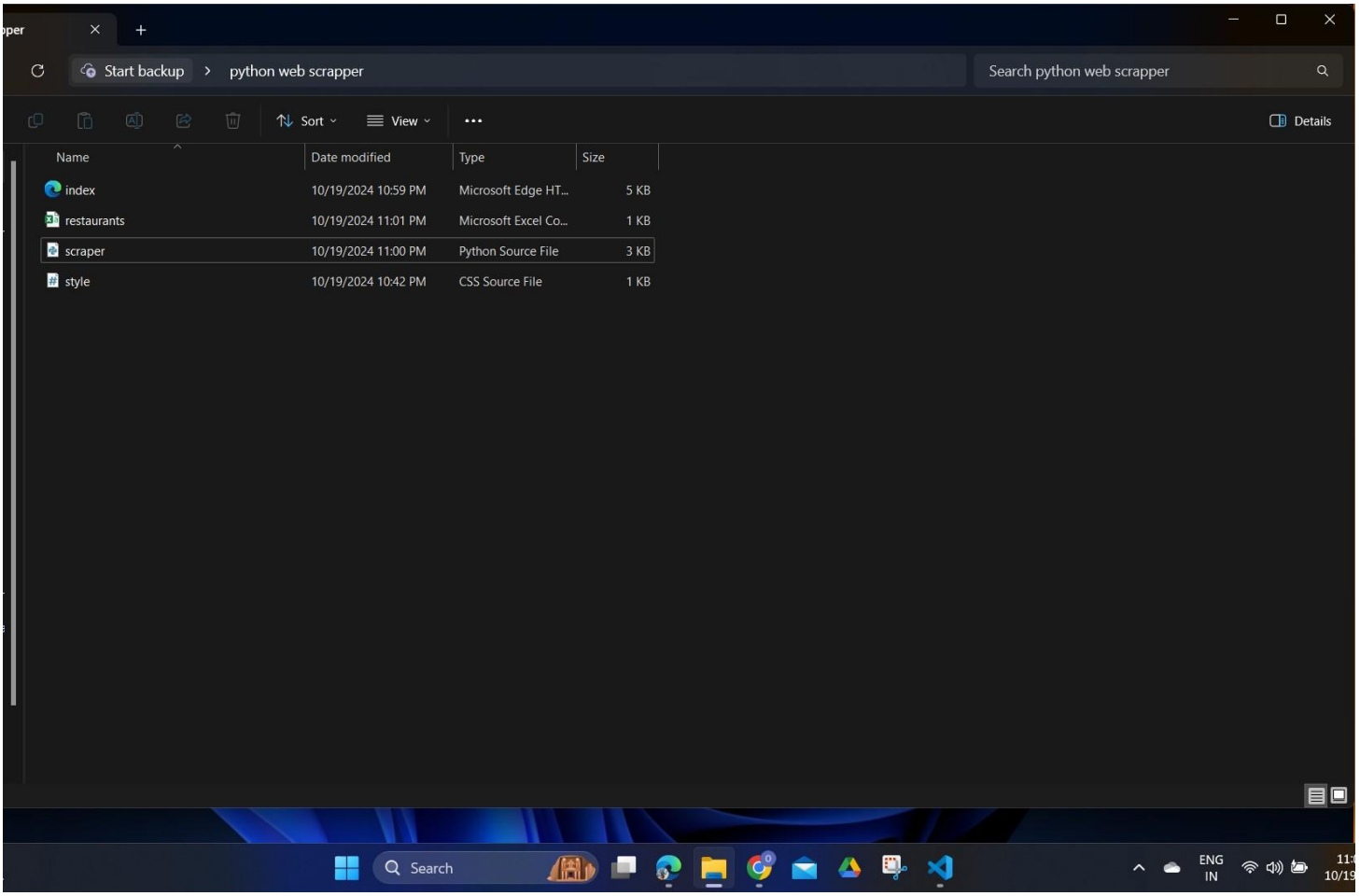
CONCLUSION

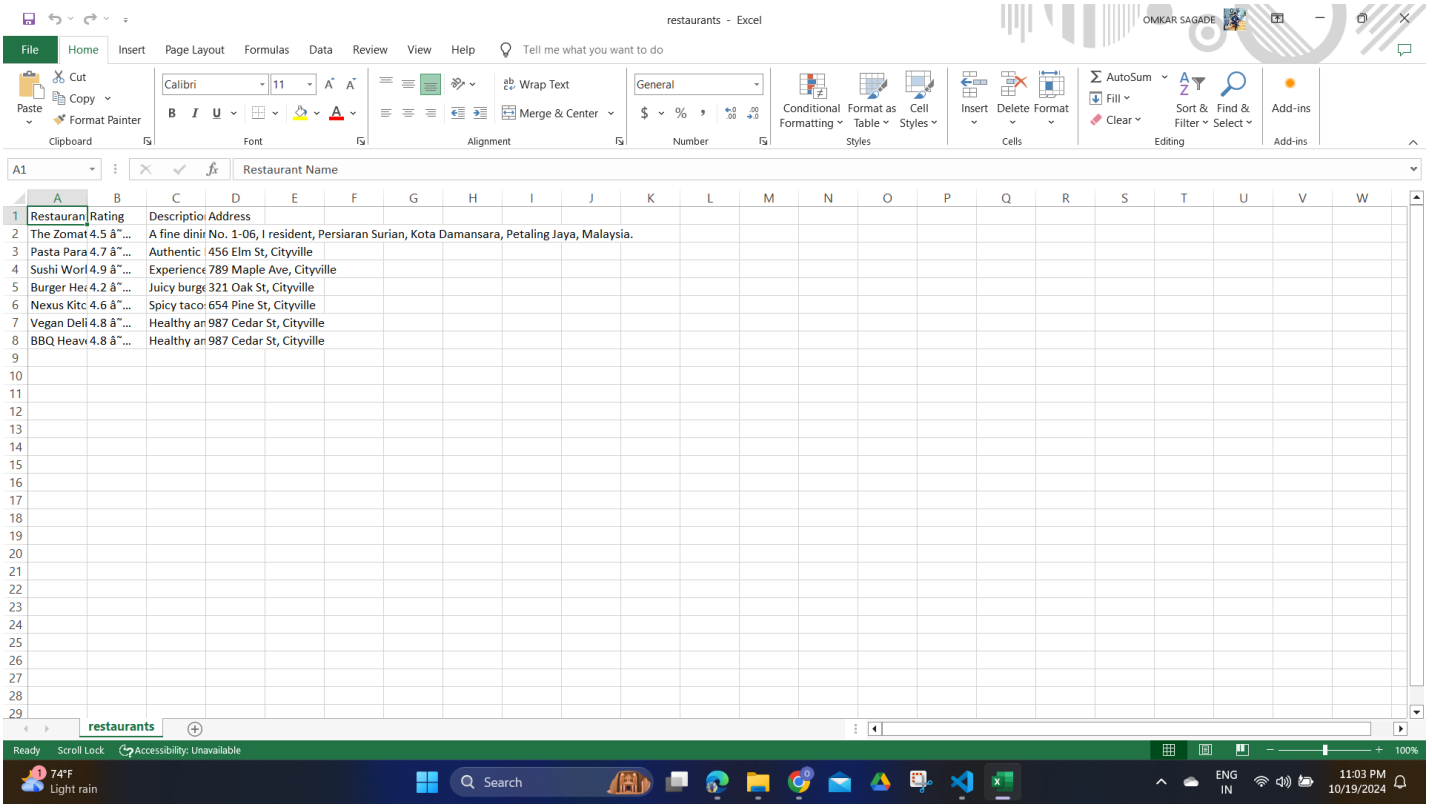
A successful example of an automated, effective method for obtaining structured data from unstructured HTML files is the Restaurant Data Scraper project. The application simplifies the process of compiling and arranging crucial restaurant information, like names, ratings, and addresses, by leveraging Python, Selenium, and BeautifulSoup. Data accessibility for additional analysis in sentiment analysis, market research, and culinary analytics is ensured by exporting the data to a CSV file. In addition to demonstrating how online scraping can be used to turn unstructured data into useful insights, this project offers a versatile framework that may be modified for comparable uses in other fields. It is a useful tool for data-driven research and decision-making for its scalability friendliness.

RESULTS AND OUTPUT :-






Start backup
python web scrapper
Search python web scrapper
Sort
View
Details
Name
Date modified
Type
Size
index
10/19/2024 10:59 PM
Microsoft Edge HT...
5 KB
restaurants
10/19/2024 11:01 PM
Microsoft Excel Co...
1 KB
scraper
10/19/2024 11:00 PM
Python Source File
3 KB
style
10/19/2024 10:42 PM
CSS Source File
1 KB
Search
ENG
IN


restaurants - Excel
OMKAR SAGADE
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Help
Tell me what you want to do
Paste
Cut
Copy
Format Painter
Clipboard
Calibri
11
Font
Wrap Text
Merge & Center
Alignment
General
Number
Conditional Formatting
Format as Table
Cell Styles
Styles
Insert
Delete
Format
Cells
AutoSum
Fill
Clear
Sort & Filter
Find & Select
Editing
Add-ins
A1
Restaurant Name
Restauran Rating Descriptio Address
The Zomat 4.5 â˜... A fine dini No. 1-06, I resident, Persiaran Surian, Kota Damansara, Petaling Jaya, Malaysia.
Pasta Para 4.7 â˜... Authentic 456 Elm St, Cityville
Sushi Worl 4.9 â˜... Experience 789 Maple Ave, Cityville
Burger Hea 4.2 â˜... Juicy burge 321 Oak St, Cityville
Nexus Kitc 4.6 â˜... Spicy taco 654 Pine St, Cityville
Vegan Deli 4.8 â˜... Healthy an 987 Cedar St, Cityville
BBQ Heav 4.8 â˜... Healthy an 987 Cedar St, Cityville
restaurants
Ready
Scroll Lock
Accessibility: Unavailable
100%
74°F
Light rain
Search
ENG
IN
11:03 PM
10/19/2024

REFERENCES

[1] A survey on Web Scraping with Python and Selenium was conducted by Sarah Fatima, Shaik Luqmaan and Nuha Abdul Rasheed in 2021 and was Published in IOSR Journal of Computer Engineering, vol. 23,Issue no. 3, pp. 01-05.

[2] In 2023, the IEEE published a comprehensive assessment of an Web Scraping, titled "Web Scraping for Data Analytics:A BeautifulSoup Implementation," written by Ayat Abodayeh, Reem Hejazi, Ward Najjar, Leena Shihadeh, Dr. Rabia Latif.

[3] In 2022, Miss. Aenugu Swetha and Dr. CH. Srihari, Dr.A.Sathyanarayana conducted a thorough analysis of web Scraping using Python for Data Analysis in the Journal of Engineering Sciences, volume 13, Issue 03,ISSN NO: 0377-9254.

[4] "Review of Web Scraping Technmiquesfor Data Analysis For Web Scraping Using Python,"Prof. Usha Nandwani and Mr.Ritesh Mishra, Mr.Amol Patil, Mr.Wasimuddin Siddiqui International Journal for Research in Enginnering Application and Management , vol.07, no. 8, pp. ISSN : 2454-9150, MAY 2021.