Report On

Real estate web scraper API

Submitted in partial fulfillment of the requirements of the Mini project in Semester VI of Third Year Artificial Intelligence & Data Science Engineering

by

Aryan Singh Rawat (Roll No. 45)
Ryan Chulliyil (Roll No. 46)
Aadil Shaikh (Roll No. 48)
Neha Singh (Roll No 56)

Under the guidance of

Prof. Sejal D'mello

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science

(A.Y. 2023-24)

Vidyavardhini's College of Engineering and Technology
Department of Artificial Intelligence & Data Science

CERTIFICATE

This is to certify that the Mini Project entitled **"Real estate web scraper API"** is a bonafide work of **Aryan Singh Rawat (Roll No. 45), Ryan Chulliyil (Roll No. 46), Aadil Shaikh (Roll No. 48), Neha Singh (Roll No 56)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in Semester VI of Third Year **"Artificial Intelligence and Data Science".**

_____
Prof. Sejal D'mello
Guide

_____	_____	_____
Ms. Sejal D'mello	Dr. Tatwadarshi Nagarhalli	Dr. H.V. Vankudre
Deputy HOD AI & DS	HOD AI &DS	Principal

Vidyavardhini's College of Engineering and Technology
Department of Artificial Intelligence & Data Science

Mini Project Approval

This Mini Project entitled "**Real estate web scraper API"** by **Aryan Singh Rawat (Roll No. 45), Ryan Chulliyil (Roll No. 46), Aadil Shaikh (Roll No. 48), Neha Singh (Roll No 56)** is approved for the degree of **Bachelor of Engineering** in Semester VI of Third Year **Artificial Intelligence and Data Science.**

Examiners

1…….....
(Internal Examiner Name & Sign)

2……
(External Examiner name & Sign)

Date:

Place:

Contents

Abstract

In the dynamic realm of real estate, the persistent obstacle of accessing comprehensive and reliable property data obstructs informed decision-making for various stakeholders, including buyers, agents, developers, and researchers [1][2]. This paper tackles this challenge by proposing a solution through the establishment of a centralized platform for aggregating real estate data. Employing Python-based web scraping techniques, our system adeptly gathers a wide array of property information, prioritizing pricing, locations, and features, while integrating stringent validation mechanisms to ensure data accuracy and dependability [3][4]. The backend infrastructure of the platform is constructed using FastAPI, facilitating seamless access to the aggregated data via an intuitive API interface, with data storage and retrieval supported by either PostgreSQL or MongoDB databases [5]. Targeting a diverse spectrum of end-users such as buyers, agents, developers, investors, and researchers, our platform serves as a valuable resource for market analysis, decision support, trend analysis, research endeavors, client assistance, app integration, and policy-making [6][7]. In addressing the research problem, this paper seeks to furnish a centralized and user-friendly solution to empower stakeholders with timely and reliable real estate data, thereby enriching decision-making processes within the competitive real estate market. The methodology employed is tailored to the task, leveraging web scraping techniques and robust validation mechanisms, with logical reasoning underpinning the analysis and argumentation, accentuating the significance of centralized data aggregation in facilitating informed decision-making in real estate. Moreover, avenues for future research are illuminated, suggesting potential directions to further advance the field and promote enhanced decision-making practices [10].

Acknowledgments

I would like to thank all people whose support and cooperation has been an invaluable asset during this Project. I would also like to thank our Guide Prof. Bhavika Gharat, for guiding me throughout this project and giving it the present shape. It would have been impossible to complete the project without his/her support, valuable suggestions, criticism, encouragement, and guidance.

I convey my gratitude to Dr. Tatwadarshi Nagarhalli, Head of Department, for his motivation and providing various facilities, which helped us greatly in the whole process of this project. I am also grateful to all other teaching and non-teaching staff members of the Artificial Intelligence and Data Science Department for directly or indirectly helping us with the completion of projects and the resources provided.

Aryan Singh Rawat (45)

Ryan Chulliyil (46)

Aadil Shaikh (48)

Neha Singh (56)

Date:

List of Abbreviations

Sr. No.	Abbreviation	Full Form
1	API	Application Programming Interface
2	DOM	Document Object Model
3	HTML	Hypertext Markup Language
4	JSON	JavaScript Object Notation
5	XML	eXtensible Markup Language
6	URL	Uniform Resource Locator
7	AWS	Amazon Web Services

1. INTRODUCTION

1.1. INTRODUCTION

The real estate industry is characterized by its dynamic nature and diverse stakeholders who heavily rely on comprehensive and reliable property data for informed decision-making. However, the vast volume and diversity of data available across online platforms pose significant challenges in efficiently aggregating and validating information. To address these challenges, this project aims to streamline real estate data collection by leveraging web scraping techniques and API integration. By utilizing Python-based web scraping and robust API infrastructure, the project endeavors to develop a centralized platform capable of aggregating, validating, and disseminating real estate data effectively. This platform will serve as a valuable resource for stakeholders across the real estate spectrum, empowering them with timely, accurate, and actionable insights for decision-making purposes.

1.2. PROBLEM STATEMENT & OBJECTIVES

Problem Statement

The real estate industry is characterized by its dynamic nature and diverse stakeholders who heavily rely on comprehensive and reliable property data for informed decision-making. However, the vast volume and diversity of data available across online platforms pose significant challenges in efficiently aggregating and validating information. To address these challenges, this project aims to streamline real estate data collection by leveraging web scraping techniques and API integration. By utilizing Python-based web scraping and robust API infrastructure, the project endeavors to develop a centralized platform capable of aggregating, validating, and disseminating real estate data effectively. This platform will serve as a valuable resource for stakeholders across the real estate spectrum, empowering them with timely, accurate, and actionable insights for decision-making purposes.

Objectives

- Develop a centralized platform for real estate data collection using web scraping and API integration.
- Streamline the aggregation and validation process of real estate information.
- Provide stakeholders with timely, accurate, and actionable insights for informed decision-making.

- Deploy the system on AWS for scalability and reliability: This objective involves deploying the Todo List project on the Amazon Web Services (AWS) cloud platform to ensure scalability, reliability, and availability.

1.3. SCOPE

The scope of this project encompasses the development of a comprehensive solution for real estate data collection, aggregation, and validation. It includes the implementation of web scraping techniques to extract pertinent information from online listings and the integration of APIs to access and manipulate data from disparate sources. The project aims to evaluate the effectiveness and efficiency of the proposed solution through experiments and statistical analyses. Additionally, it explores the potential benefits of centralized data aggregation for stakeholders within the real estate industry. Through collaboration and innovation, the project strives to contribute to the advancement of the real estate ecosystem by enhancing decision-making processes and empowering stakeholders with actionable insights.

2. LITERATURE SURVEY

In this section, we conduct a comprehensive review of existing literature pertaining to real estate web scraping and Application Programming Interfaces (APIs). We delve into previous research studies, methodologies employed, and key findings to provide insights into the current state of knowledge in this domain.

2.1. SURVEY OF EXISTING SYSTEM

Predictive Modeling for Real Estate Market Analysis: Numerous studies have delved into predictive modeling techniques to analyze real estate market trends and property prices [8][6]. By leveraging web scraping methods to gather data from diverse real estate websites, researchers have developed predictive models capable of forecasting market fluctuations and property valuations [9]. The integration of APIs for data retrieval has streamlined the process, allowing for the collection of real-time information to inform decisionmaking [1].

Efficiency and Accuracy in Web Scraping: The optimization of web scraping processes has been a focal point in recent research [3]. Studies have explored various techniques, including DOM tree construction and string-based extraction methods, to improve the efficiency and accuracy of data extraction [3]. While traditional DOM-based approaches provide structural insights into web pages, string-based methods offer faster extraction speeds, making them suitable for large-scale data collection tasks [3]. Additionally, the utilization of additional information, such as tag repetition and inner tag counts, has further enhanced the efficiency of web scraping algorithms [3].

Utilization of Nontraditional Data Sources: Some studies have turned to nontraditional data sources, such as Craigslist listings, to analyze rental housing markets [9]. By collecting, cleaning, and analyzing vast quantities of rental housing data, researchers have uncovered nuanced spatial and temporal patterns within metropolitan housing markets [9]. These insights offer real-time snapshots of rental market dynamics, complementing traditional census data sources and providing valuable information for policymakers, investors, and tenants alike [9].

2.2. LIMITATION IN EXISTING SYSTEM OR RESEARCH GAP

Scalability and Generalization: While existing studies have demonstrated the effectiveness of predictive modeling in real estate market analysis, there remains a need to assess the scalability and generalization of these models [8].

Evaluating model performance across diverse geographic regions and market conditions would enhance their applicability and reliability for stakeholders [8]. Optimization of Web Scraping Techniques: Despite advancements in web scraping methodologies, opportunities exist for further optimization [3]. Future research could explore innovative approaches and technologies to enhance the speed, accuracy, and robustness of data extraction processes, ensuring the timely retrieval of high-quality information from real estate websites [3].

Integration of Multisource Data: Integrating data from multiple sources remains a challenge in real estate research. While studies leveraging nontraditional data sources have provided valuable insights, combining data from disparate platforms poses technical and methodological complexities [9]. Addressing these challenges could unlock opportunities for comprehensive and real-time analyses of real estate dynamics, benefiting researchers, practitioners, and policymakers alike [9].

2.3. MINI PROJECT CONTRIBUTION

Implementation of Scraping Logic: You could contribute by implementing the actual scraping logic for various real estate websites. This involves writing code to handle requests, parse HTML content, extract relevant data points like price, location, and property details, and handle pagination [9].

Refinement of Parsing Logic: You could improve the parsing logic to make it more robust and efficient. This might involve handling edge cases, optimizing selectors for HTML elements, and improving error handling [3].

File Handling and Serialization: Contributing to the optional step of saving scraped data to a file involves implementing functionality to write data in various formats like JSON, CSV, or XML. You could also improve file handling capabilities, such as appending data to existing files or managing large datasets efficiently [7].

Pagination Handling: You could work on improving pagination handling to make it more flexible and reliable across different real estate websites. This might involve detecting pagination patterns automatically or implementing custom pagination strategies for specific websites[2,8].

Error Handling and Logging: Implementing robust error handling mechanisms and logging capabilities would improve the reliability and maintainability of the API. Users should be informed of any errors that occur during the scraping process, along with relevant details for troubleshooting [5,8].

Deployment on AWS: Ensure seamless deployment of the platform on AWS infrastructure for scalability, reliability, and security. Utilize AWS services such as EC2 for hosting, RDS or DynamoDB for database management, S3 for file storage, and CloudWatch for monitoring and logging. Implement auto-scaling and load balancing to handle varying traffic loads effectively. Additionally, incorporate AWS security best practices to safeguard data and infrastructure.

Data Visualization: Implement advanced data visualization techniques to present real estate market trends, pricing dynamics, and property analytics through interactive charts, graphs, and dashboards.

Property Comparison Tool: Develop a property comparison tool that allows users to compare multiple properties side by side based on various parameters such as price, location, size, amenities, and more.

3. PROPOSED SYSTEM

3.1. ARCHITECHTURE/FRAMEWORK/BLOCK DIAGRAM

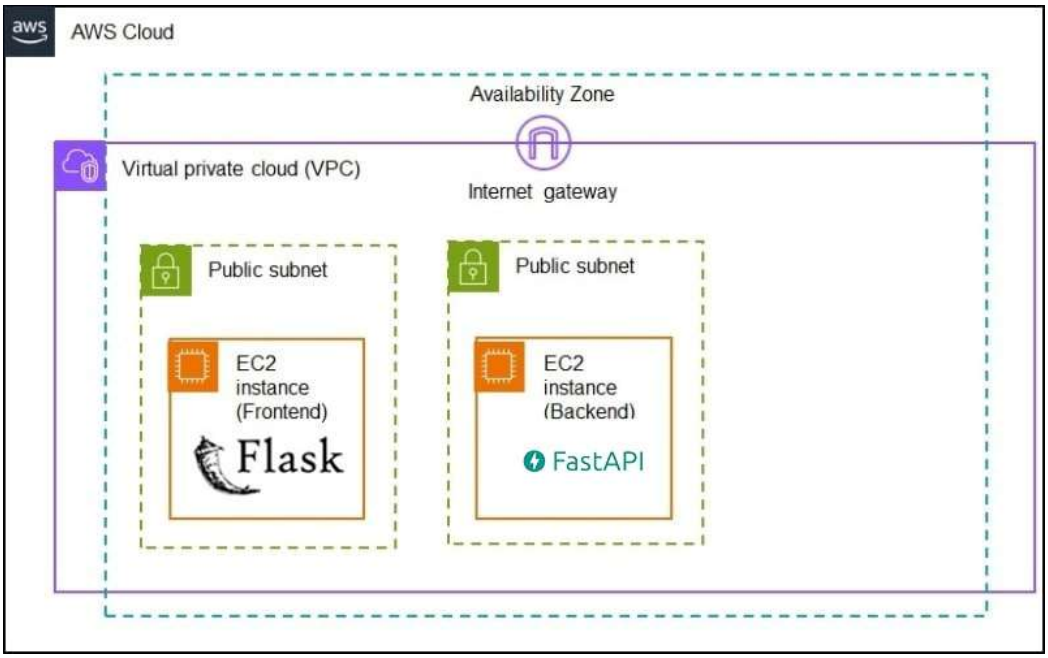

Figure 1: AWS architechture

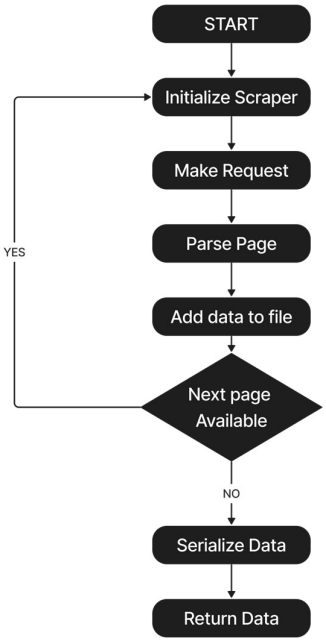

Figure 2: Flowchart web scraper

3.2. ALGORITHM AND PROCESS DESIGN

1. **Initialize Scraper:** This step likely involves setting up the API and any necessary parameters for the scraping job. This might include things like the target URL, what data to extract, and how to handle pagination.

2. **Make Request:** The API sends a request to the target real estate website to retrieve the HTML content of the listing page.
3. **Parse Page:** Once the HTML content is received, the API parses it to extract the relevant data points. This typically involves using a parsing library to identify the HTML elements that contain the desired information, such as price, location, number of bedrooms, and property description.
4. **Add data to file:** This step might involve saving the scraped data to a file for later use. The file format could be JSON, CSV, or another format depending on the needs of the application.
5. **Next Page Available:** The API checks if there are additional pages of listings to scrape. This might involve looking for a "next page" button or link on the current page.
    - **No:** If there are no more pages, the process moves to step 7.
    - **Yes:** The process loops back to step 2 to make a request for the next page of listings.
6. **Serialize Data:** This step involves formatting the extracted data into a structured format, such as JSON or CSV, for easier storage and use.
7. **Return Data:** The API returns the scraped and formatted data to the user.

Fast API is a Python framework for building web APIs. It is designed to be high-performing and easy to use. In the context of real estate web scraping, Fast API would provide the structure for the API itself, handling things like routing and request handling.

Here are some of the benefits of using a web scraping API for real estate data:

- **Efficiency:** Scraping data manually can be time-consuming and error-prone. A web scraping API can automate the process and extract data much faster.
- **Accuracy:** Web scraping APIs can be designed to be very accurate, especially when used with well-maintained websites that have consistent HTML structures.
- **Scalability:** Web scraping APIs can be easily scaled up or down to meet the needs of your project.

3.3. DETAILS OF HARDWARE AND SOFTWARE

Hardware Details: The hardware required to run the algorithm will depend on the scale of the scraping operation. For a small scraping operation, a standard computer with a decent internet connection would suffice. For larger scraping operations, a more powerful computer or server may be required.

Software Details: The software required to run the algorithm will depend on the specific implementation. However, some general purpose software tools that could be used include:

- Python: Python is a popular programming language that is often used for web scraping. There are a number of Python libraries available that can be used for web scraping, such as Beautiful Soup and Scrapy.
- Fast API: Fast API is a Python framework for building web APIs. It can be used to create a web scraping API that can be accessed by other applications.
- Web scraping library: There are a number of web scraping libraries available in Python, such as Beautiful Soup and Scrapy. These libraries can be used to parse HTML content and extract data.
- AWS (Amazon Web Services): AWS provides a wide range of cloud computing services that can be leveraged for deploying, hosting, and scaling web scraping applications. Services such as EC2, RDS, S3, and CloudWatch can be utilized for infrastructure, database management, file storage, and monitoring, respectively.

3.4. EXPERIMENT AND RESULT FOR VALIDATION AND VERIFICATION

Output

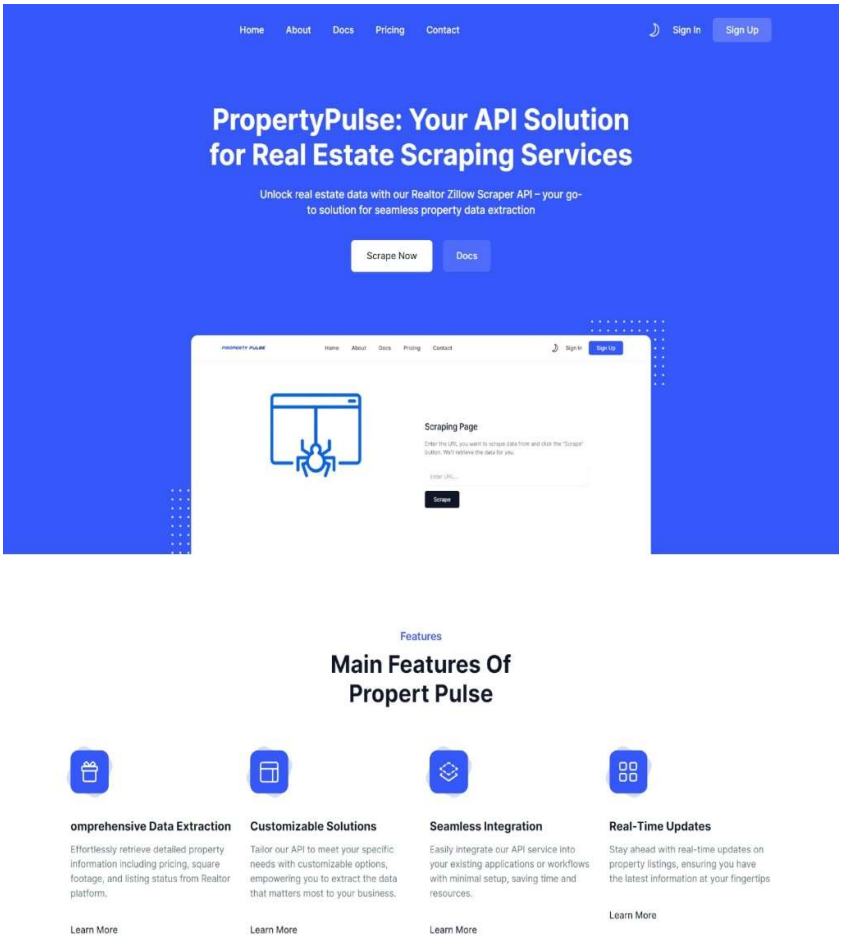

Figure 3: Home page

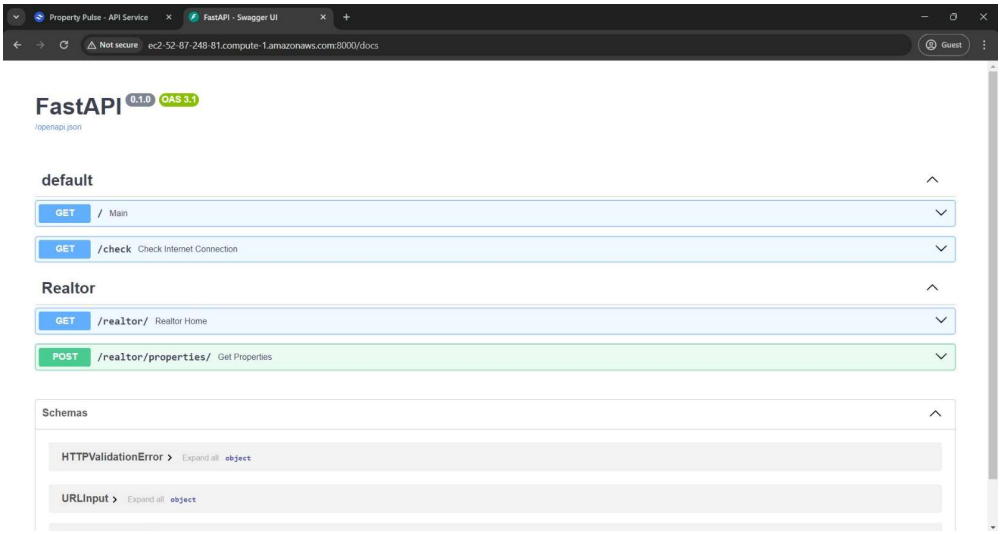

Figure 4: Background API Calls

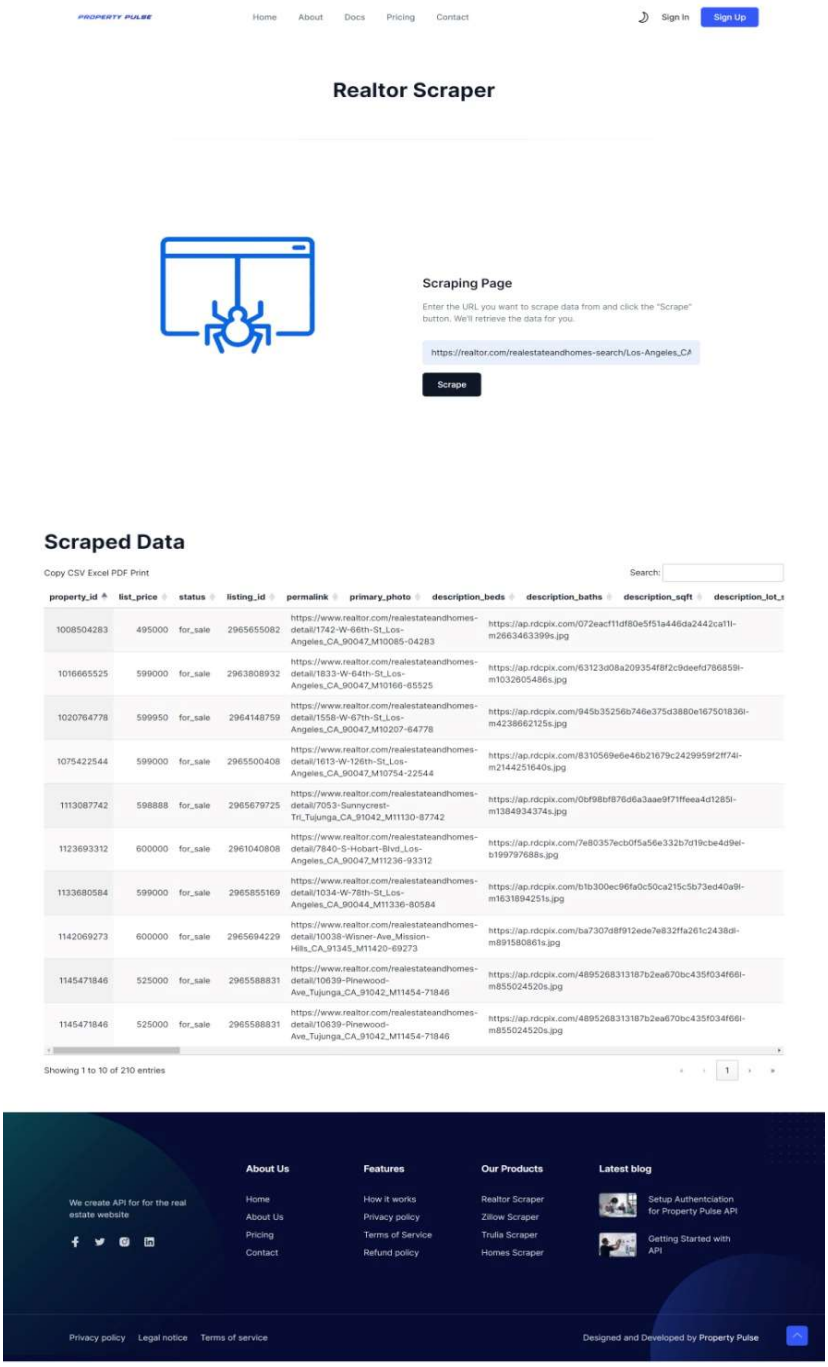

Figure 5: Output Screen

Figure 6: Response JSON

Parameters:

- property_id: Unique identifier for the property.
- list_price: Price of the property.
- search_promotions: Promotions related to the property search (null in this case).
- primary_photo: URL of the primary photo of the property.
- rent_to_own: Information about rent-to-own option (null in this case).
- listing_id: Unique identifier for the listing.
- matterport: Indicates whether Matterport virtual tour is available (false in this case).
- virtual_tours: URLs for virtual tours (null in this case).
- status: Current status of the property (for sale in this case).
- products: Information about products related to the property, such as agent and broker.
- source: Information about the source of the listing, including ID, type, and agent details.
- lead_attributes: Attributes related to leads and contacts, including lead type and mortgage information.
- community: Information about the community (null in this case).
- permalink: Permalink to view the property online.
- price_reduced_amount: Amount of price reduction (if any).
- : Details about the property, such as number of bedrooms, bathrooms, square footage, and type.
- location: Information about the location of the property, including address, coordinates, and county.
- open_houses: Details about any open houses (null in this case).

- branding: Information about branding, such as office name and photo.
- flags: Flags indicating various statuses of the listing, such as whether it's a new listing or contingent.
- list_date: Date when the property was listed.
- photos: URLs of additional photos of the property.
- advertisers: Information about advertisers, such as seller type.

Resulting Data:

The resulting data encapsulates crucial information about the listed property. This includes the property ID, list price, status (whether it's for sale), details about the listing source (such as the real estate agency), property description (including number of beds, baths, square footage, etc.), location details (address, coordinates, county), and various flags indicating specific attributes of the listing (such as whether it's contingent or a new listing).

The response also includes URLs to view the primary photo of the property and a street view of its location. Additionally, it provides information about any advertisers associated with the listing, such as the type of advertiser (seller, builder).

3.5. ANALYSIS

The parameters outlined in the response JSON are essential for understanding and analyzing the characteristics of the listed property. They provide a structured format for accessing key details such as price, location, and property features. These parameters enable efficient processing and utilization of the data within applications or analytical tools [8].

The resulting data, obtained through the parameters, offers a comprehensive overview of the property listing. It contains pertinent details required for potential buyers, real estate agents, or analytical systems to evaluate the property's suitability and make informed decisions [5,7].

Overall, by providing structured parameters and resulting data, the response facilitates seamless integration with various real estate applications, enabling efficient access to and utilization of property listing information [3,9].

3.6. CONCLUSION AND FUTURE WORK

Conclusion: The project sets out to address the challenges faced by the real estate industry in aggregating and validating vast volumes of data from online platforms. By leveraging web scraping techniques and API integration, the aim is to develop a centralized platform capable of efficiently collecting, validating, and disseminating

real estate data. This initiative is driven by the recognition of the critical role data plays in informing decision-making across various stakeholders within the industry.

Through the development of this platform, stakeholders will benefit from timely, accurate, and actionable insights, empowering them to make informed decisions. By streamlining the aggregation and validation process, the project seeks to enhance the efficiency and effectiveness of real estate data utilization.

Future Work: Moving forward, there are several areas for potential expansion and improvement:

1. Enhanced Data Quality Assurance: Continuously refining the data validation processes to ensure the accuracy and reliability of the collected real estate data. Implementing robust quality assurance measures will further enhance the credibility of the platform among stakeholders.
2. Advanced Analytics Capabilities: Introducing advanced analytics features such as predictive modeling, trend analysis, and market forecasting to provide stakeholders with deeper insights into real estate trends and dynamics. This will enable users to anticipate market shifts and make proactive decisions.
3. Integration with Emerging Technologies: Exploring opportunities to integrate emerging technologies such as machine learning, natural language processing, and computer vision to automate data extraction, improve data categorization, and enhance user experience.
4. Expansion of Data Sources: Expanding the scope of data collection to include additional sources beyond online listings, such as social media, government databases, and proprietary datasets. Diversifying data sources will enrich the platform's dataset and provide users with a more comprehensive understanding of the real estate landscape.
5. User Feedback and Iterative Development: Soliciting feedback from users and stakeholders to identify pain points, gather requirements, and prioritize feature development. Adopting an iterative development approach will ensure that the platform evolves to meet the evolving needs of its users.

4. REFERECE

[1] Adrien Barbaresi. 2021. Trafilatura: A web scraping library and command-line tool for text discovery and extraction. In Proceedings of ACL

[2] Daniel Bauer, Judith Degen, Xiaoye Deng, Priska Herger, Jan Gasthaus, Eugenie Giesbrecht, Lina Jansen, Christin Kalina, Thorben Kräger, Robert Märtin, et al. 2007. Fiasco: Filtering the internet by automatic subtree classification, osnabruck. In Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop, incorporating Clean Eval

[3] Herman, M. 2012. "Scraping Web Pages with Scrapy." Michael Herman, November 5. http://mherman.org/blog/2012/11/05/scraping-web-pages-with-scrapy/.

[4] Pollock, K. 2016. "Policy: Urban Physics." Nature 531 (7594): S64–S66. doi:10.1038/531S64a.

[5] Quigley, J., and S. Raphael. 2004. "Is Housing Unaffordable? Why Isn't It More Affordable?" Journal of Economic Perspectives.

[6] Jean-Charles Bricongne, Baptiste Meunier, Sylvain Pouget 2023. "Web-scraping housing prices in real-time: The Covid-19 crisis in the UK" Journal of Housing Economics

[7] C. H. Yang, B. Lee, and Y. Da Lin, "Effect of Money Supply, Population, and Rent on Real Estate: A Clustering Analysis in Taiwan" 2022, doi: 10.3390/ math10071155.

[8] Ali Can ÜZÜMCÜ, Nazmiye ELİGÜZEL, "Predictive Analysis Using Web Scraping for the Real Estate Market in Gaziantep"

[9] Geoff Boeing, Paul Waddell, "New Insights into Rental Housing Markets across the United States: Web Scraping and Analyzing Craigslist Rental Listings" Journal of Planning Education and Research 1–20, 2016

[10] Moaiad Ahmad Khder, "Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application" Int. J. Advance Soft Compu. Appl, Vol. 13, No. 3