WEB DATA SCRAPING AND DATA ANALYSIS IN E-COMMERCE WEBSITE

A PROJECT REPORT

Submitted by

KUNGUMA ROSHAN M [REGISTER NO 211421104242]

PARTHIBAN S [REGISTER NO 211421104915]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2025

BONAFIDE CERTIFICATE

Certified that this project report **"WEB DATA SCRAPING AND DATA ANALYSIS IN E-COMMERCE WEBSITE"** is the bonafide work of **"KUNGUMA ROSHAN M (211421104242),PARTHIBAN S (211421104915)"** who carried out the project work under my supervision.

SIGNATURE

P. DEEPA, M.E(Ph.D),

ASSOCIATEPROFESSOR

DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE ,

NASARATHPETTAI,

POONAMALLEE,

HENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voice Examination held on .....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **KUNGUMA ROSHAN M (211421104242), PARTHIBAN(211422104915) hereby declare that this project report titled** 'WEB DATA SCRAPING AND DATA ANALYSIS IN E-COMMERCE WEBSITE' under the **guidance of K.CINTHUJHA, M.E Assistant Professor is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.**

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR, M.E,Ph.D.,** and **Dr. SARANYASREE SAKTHI KUMAR B.E,MBA,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K. Mani, M.E,Ph.D.,** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L. JABASHEELA , M.E,Ph.D.,** for the support extended throughout the project.

We would like to thank our **Project Coordinator K. CINTHUJA, M.E** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

KUNGUMA ROSHAN M

PARTHIBAN S

ABSTRACT

The "**WEB DATA SCRAPING AND DATA ANALYSIS IN E-COMMERCE WEBSITE**" is an innovative use of web scraping techniques, important data is extracted from e-commerce platforms. The acquired data is then analyzed to provide insights into consumer behavior, market trends, and rival plans. A range of data analysis techniques, such as sentiment analysis, predictive modeling, and descriptive statistics, are utilized to derive actionable insights for e-commerce enterprises. In the cutthroat world of e-commerce, the findings help to improve decision-making procedures, maximize marketing tactics, and boost overall business success.

TABLE OF CONTENT

INTRODUCTION

1.1 OVERVIEW

This project aims to develop an automated web scraping solution using **UiPath** to extract product data from e-commerce websites based on user-defined conditions. Users can set filters such as price range, brand, discount percentage, and customer ratings, and the bot will collect relevant product details, including names, prices, availability, and reviews. **UiPath automation capabilities**, including UI automation, data scraping, and API integrations, will be leveraged to handle both static and dynamic web pages efficiently. To bypass website restrictions like CAPTCHAs and dynamic content loading, the bot will use **headless browsing, delay mechanisms, and API-based extraction (where possible)**. Extracted data will be structured and stored in **Excel, CSV, databases, or dashboards**, allowing users to analyze price trends, compare products, and track competitor pricing. The project ensures compliance with ethical and legal guidelines while enhancing decision-making for businesses and consumers.

1.2 PROBLEM DEFINITION

Manually collecting product data from e-commerce websites is time-consuming and inefficient. Websites use dynamic content and anti-scraping measures like CAPTCHAs and IP blocking, making data extraction difficult. Traditional methods fail to handle these challenges effectively. This project develops an automated web scraping system to extract data based on user-defined conditions while ensuring ethical compliance.

2. LITERRATURE SURVEY

Web scraping plays a crucial role in e-commerce for extracting product details, price comparison, and market analysis. Traditional web scraping techniques use programming languages like **Python** with libraries such as **BeautifulSoup, Scrapy, and Selenium**. However, these methods require coding expertise and often struggle with **dynamic content, JavaScript loading, and anti-scraping mechanisms**. **UiPath**, a leading **Robotic Process Automation (RPA)** tool, provides a **no-code alternative** for automating web data extraction efficiently. Its **Data Scraping Wizard** allows easy extraction of structured data, while **UI automation handles dynamic websites** effectively. Research indicates that **RPA-based web scraping is more flexible and user-friendly** compared to traditional coding-based methods. UiPath supports **AI-powered automation** to bypass CAPTCHAs, handle pop-ups, and interact with JavaScript-based elements. Additionally, **API integration** can be used as a legal and ethical alternative when direct scraping is restricted. Studies suggest that combining **machine learning with RPA** enhances adaptability to changing website structures, reducing maintenance efforts. Despite its advantages, **ethical and legal concerns** remain critical, as unauthorized scraping may violate website terms of service. Businesses must ensure compliance by following ethical guidelines and leveraging **public APIs where available**. This project utilizes UiPath's capabilities to extract e-commerce data based on **user-defined conditions such as price, brand, and ratings**. The scraped data is structured into **Excel, CSV, or databases** for easy analysis and reporting. This approach ensures a **scalable, efficient, and legally compliant** solution for automated web scraping in e-commerce.

3. SYSTEM ANALYSIS

3.1EXISTING SYSTEM

The Existing systems use UiPath **for web scraping in various industries. For example, businesses in** e-commerce **use UiPath to automate** price comparison **across multiple websites by extracting product data like prices and availability.** Market intelligence **systems track competitor pricing and promotions using UiPath's scraping capabilities.** Lead generation **is another use case, where UiPath scrapes contact details from directories or social media. In** real estate**, UiPath is used to collect property listings and analyze market trends. Other applications include** job listing aggregation **and** data collection for research**. UiPath's** Data Scraping Wizard **and** UI Automation **features are key tools, along with** OCR **for extracting data from images. The platform allows businesses to automate data extraction efficiently, ensuring quick, accurate, and structured data for analysis.**

3.2 PROPOSED SYSTEM

Proposed systems using **UiPath** include automating **dynamic content scraping** from JavaScript websites through **UI automation**. Another system focuses on **competitor price monitoring**, extracting product data across e-commerce platforms for analysis. A **lead generation system** scrapes contact information from directories and social media for sales. A **real-time market intelligence system** collects data from news and competitor websites. Additionally, a **legal compliance framework** ensures scraping follows **robots.txt** and ethical guidelines. These systems aim to enhance efficiency, scalability, and legal compliance in web scraping tasks.

3.3 HARDWARE ENVIROMENT

1. Processor(CPU)
2. RAM
3. Storage
4. Network

3.4 SOFTWARE ENVIROMENT

1. UI-Path Studio
2. Web Brower
3. Database Storage

4. SYSTEM DESIGN

4.1 Flow Chart

The web scraping process using UiPath starts by launching a browser and navigating to the target e-commerce website. The bot then identifies and extracts relevant product details such as names, prices, and availability using UiPath's data scraping capabilities. If the website has multiple pages, the bot loops through them to ensure all data is collected efficiently. Once the extraction is complete, the data is stored in a structured format like an Excel file, database, or CSV. Finally, the browser is closed, and the automation process ends, ensuring efficient and accurate data retrieval.

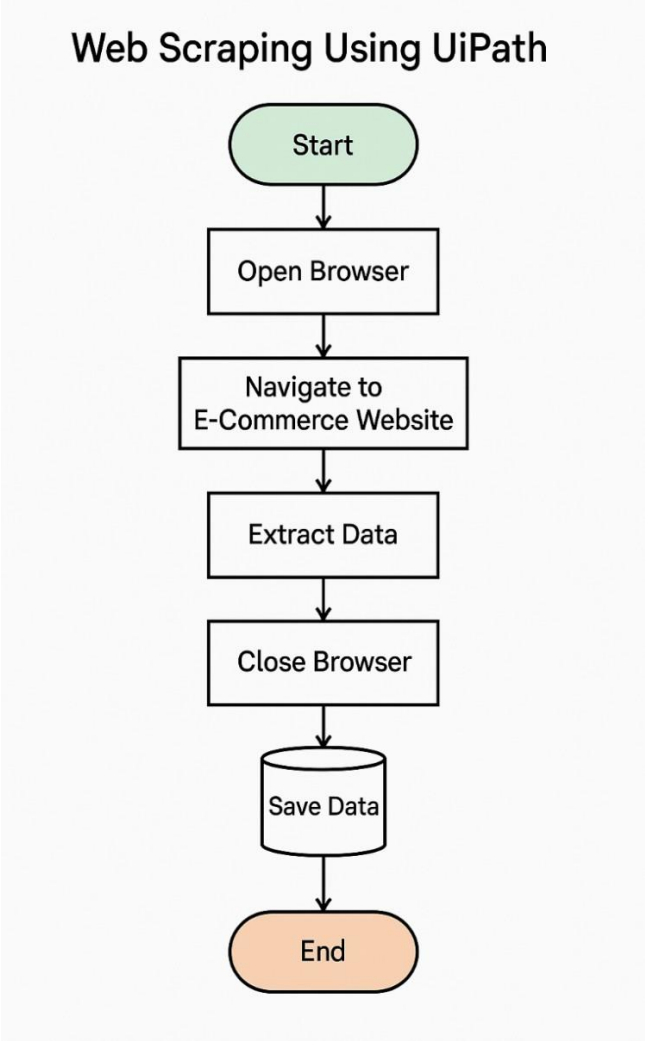


Fig 4.1 Flow-chart

4.2 Architecture Diagram

The architecture diagram represents the workflow of UiPath automation integrated with a web application. On the client side, users interact with the system through a browser-based web application built with HTML and JavaScript. This web application communicates with UiPath components, including UiPath Agent, which manages execution through UiPath Studio and the Executor. The OData REST API endpoints act as the bridge between the front end and backend, handling configuration, logging, monitoring, and queue management. The service layer processes business logic and automation tasks, ensuring efficient execution. Then the data will store it in excel file .

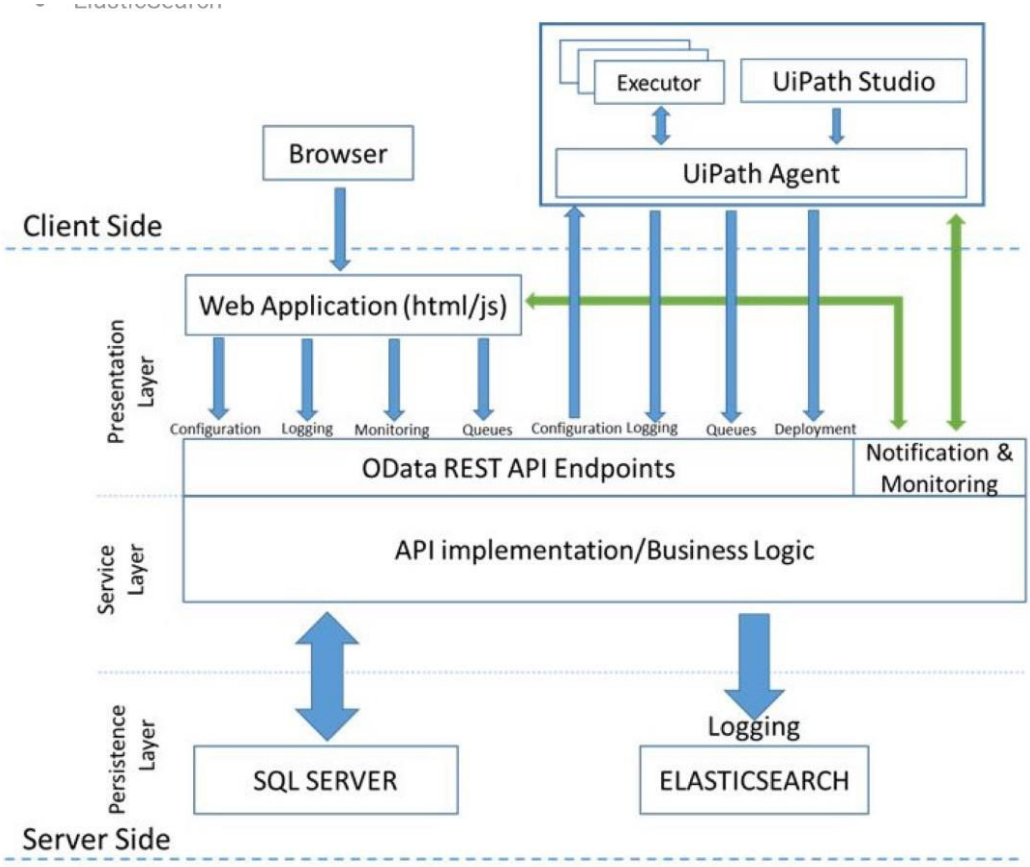


Fig 4.2 Architecture Diagram

4.3 Data Flow Diagram

4.3.1 Zero level DFD

This for web scraping an e-commerce website represents the overall process at a high level. The user initiates the process by providing the target website URL and specifying the data fields to extract. The UiPath bot then navigates to the website, scrapes the required product information, and processes the extracted data to ensure it is structured and clean. Once processed, the data is stored in a database, Excel, or CSV file for further analysis and reporting. Finally, the user retrieves the stored data for insights, automation, or business decision-making.

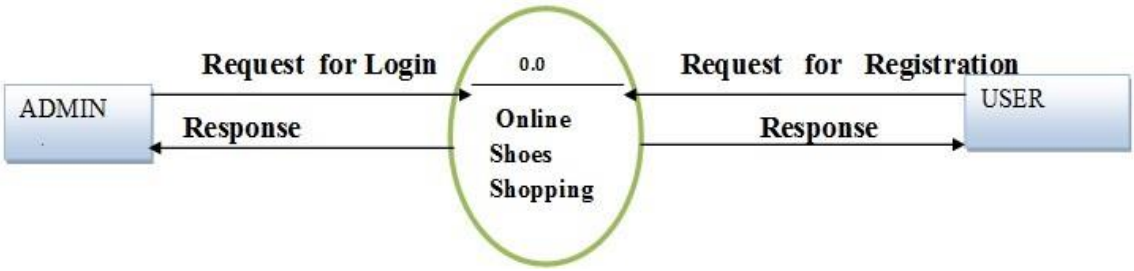


Fig 4.3.1 0 level DFD

4.3.2 First Level DFD

The **Level 1 Data Flow Diagram (DFD)** for web scraping an e-commerce website provides a more detailed breakdown of the process. First, the user inputs the website URL and specifies the data fields to be extracted, such as product name, price, and availability. Then store it in the excel sheet

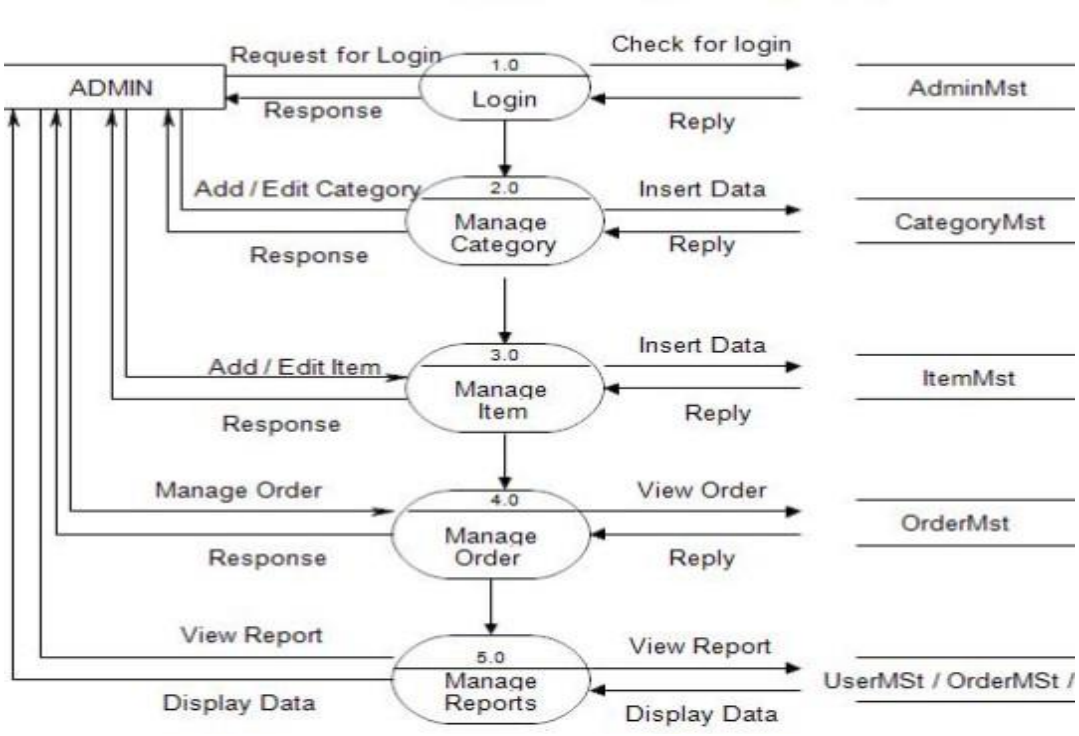


Fig 4.3.2 First Level DFD

4.3.3 Second Level DFD

Web scraping is the automated process of extracting data from websites by sending requests and retrieving their content. It typically involves downloading the HTML or other data formats from a target site. The retrieved content is then parsed to extract specific information like text, images, or links. This data is often cleaned and organized into structured formats for easier analysis. Web scraping is widely used in industries for gathering large datasets from the web for research, business intelligence, or competitive analysis.

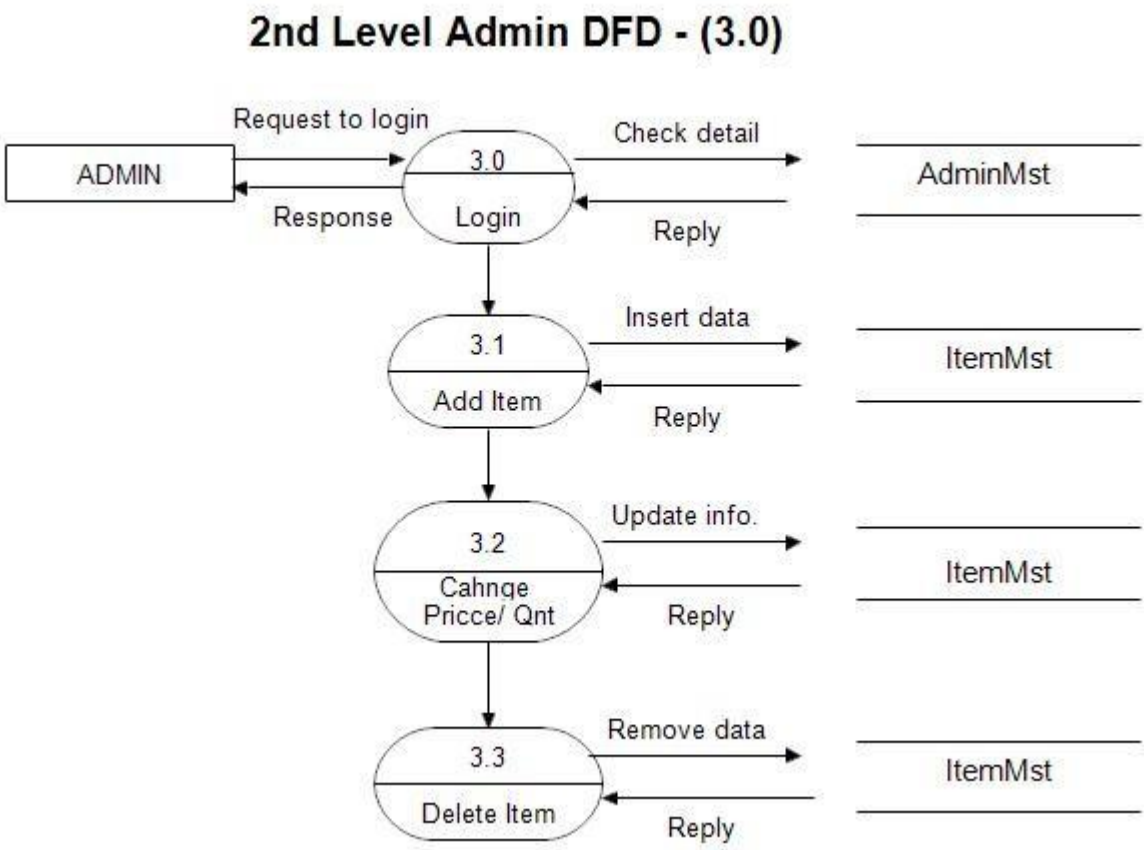


Fig 4.3.3 Second Level DFD

4.4 Use Case Diagram

A use case diagram for web scraping illustrates the interactions between the user and the system, focusing on the key tasks involved. The primary actor, the **User**, initiates the web scraping process by providing target URLs or search queries. The **Web Scraping System** performs tasks like sending HTTP requests, retrieving web content, and parsing data. Additional use cases include **Data Extraction**, where the relevant information is identified and parsed, and **Data Output**, which delivers the structured results to the user. This diagram highlights the functional requirements and user-system interactions in the web scraping process.

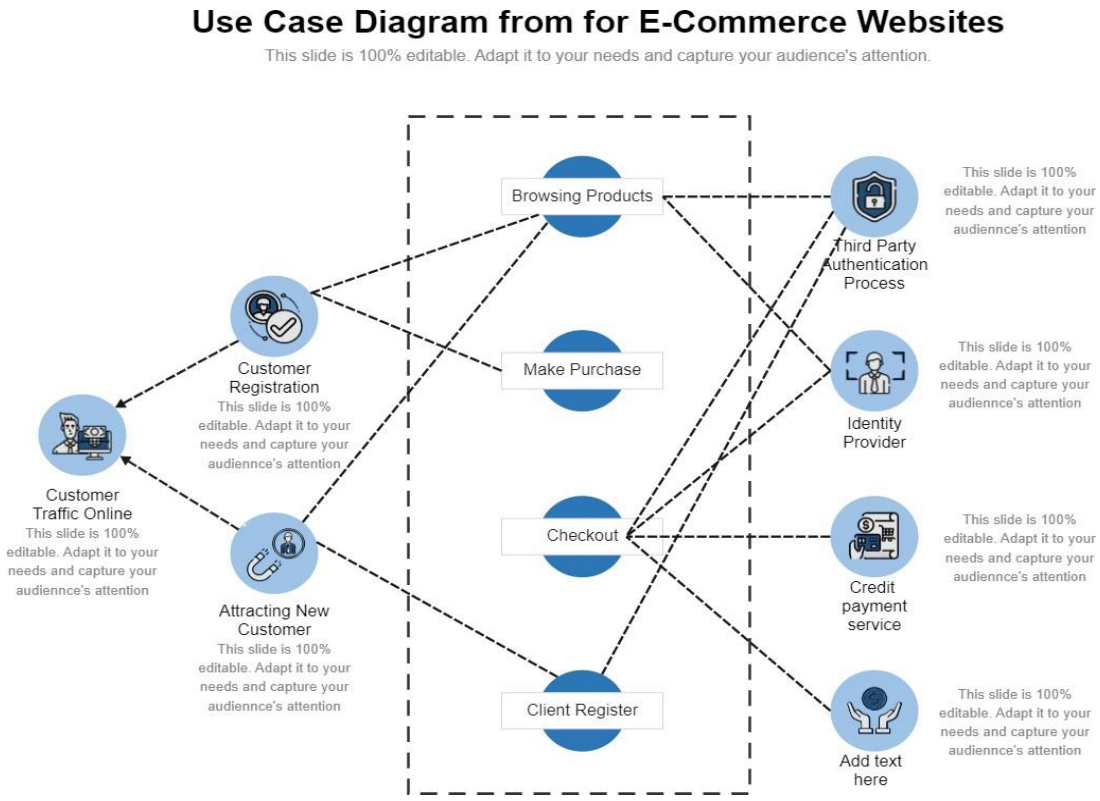


Fig 4.4 Use Case Diagram

4.5 Sequence Diagram

A sequence diagram for web scraping illustrates the order of interactions between the user, the web scraping system, and the target website. The user starts by sending a request with the target URL to the web scraping system. The system then sends an HTTP request to the target website to fetch the raw data. The website responds with the data, which the system processes by extracting the relevant information. Finally, the processed data is sent back to the user in a structured format, completing the sequence of actions in the web scraping process.

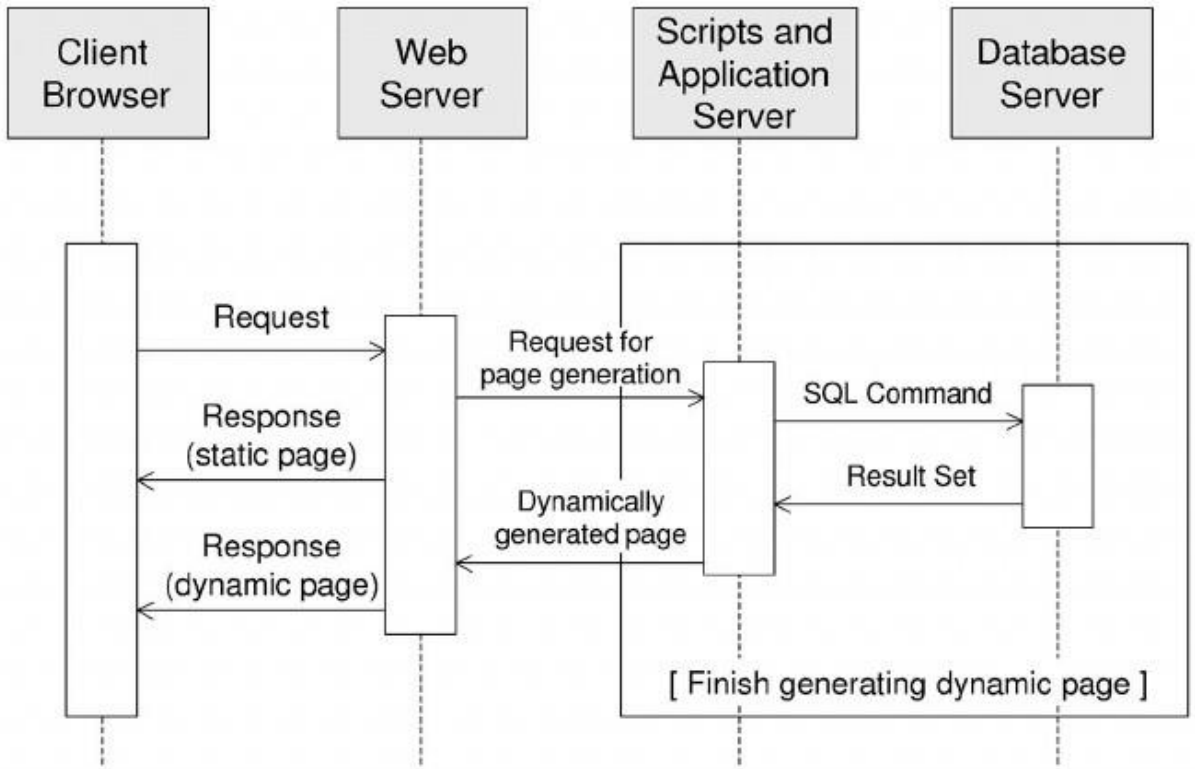


Fig 4. 5 Sequence Diagram

4.6 Activity Diagram

An activity diagram for web scraping outlines the flow of actions in the process. It begins when the user provides the target URL and parameters, prompting the system to send an HTTP request to the website. The system receives the raw data and proceeds to parse and extract relevant information. The extracted data is then cleaned, structured, and formatted to meet user requirements. Finally, the processed data is delivered to the user, completing the web scraping process.

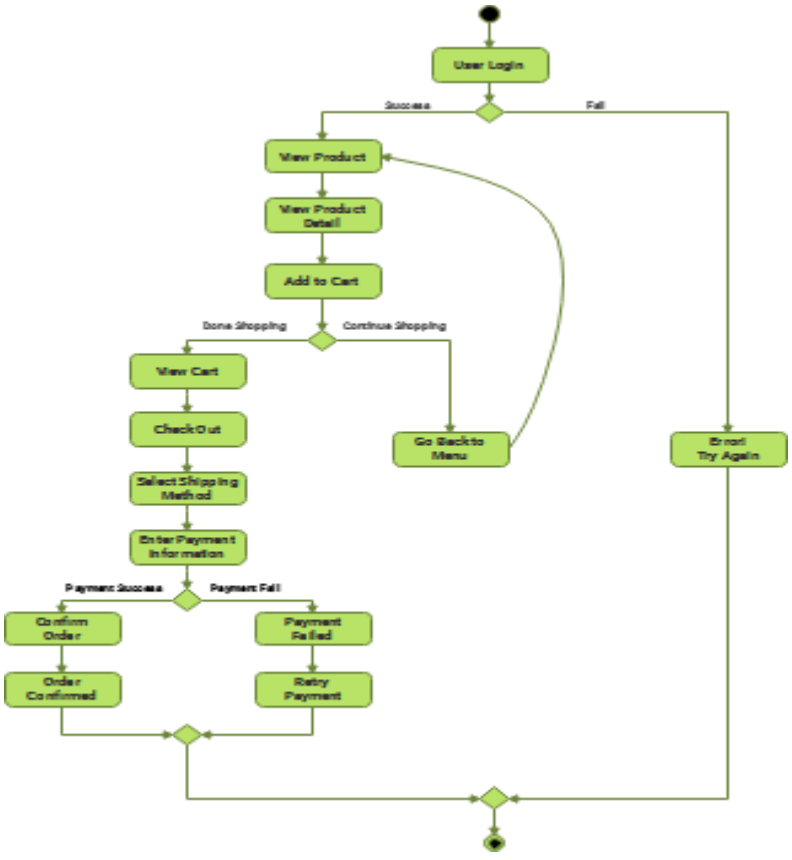


Fig 4.6 Activity Diagram

4.7 Class Diagram

A class diagram for web scraping outlines the key classes and their relationships involved in the process. The primary classes could include **WebScraper**, responsible for managing the scraping logic, **HttpRequest**, for handling requests to the target website, and **DataParser**, which extracts and processes the data from the raw content. Additionally, a **DataStorage** class may be used to store the structured data. Relationships between these classes are shown, such as **WebScraper** interacting with **HttpRequest** and **DataParser** to complete the scraping process. This diagram helps visualize the system's structure and the interactions between its components.

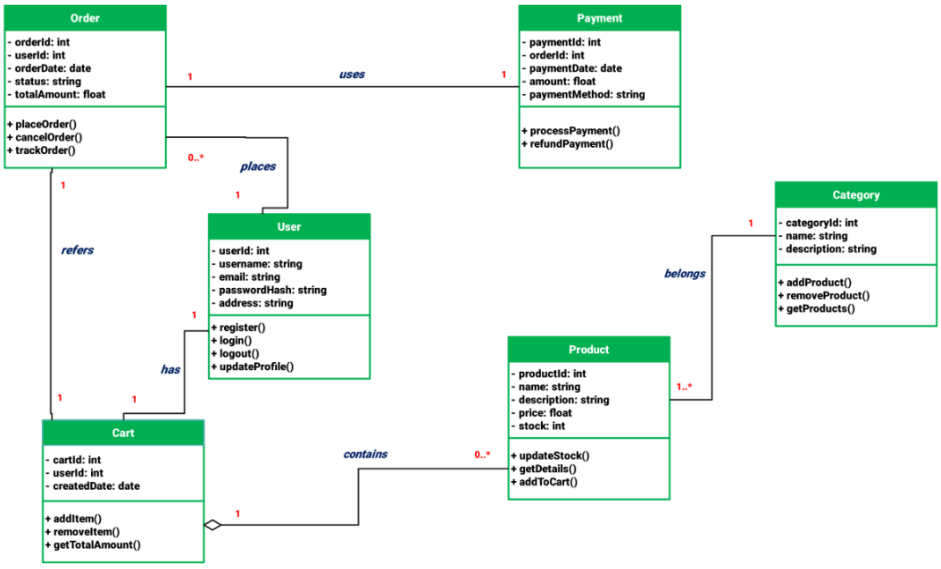
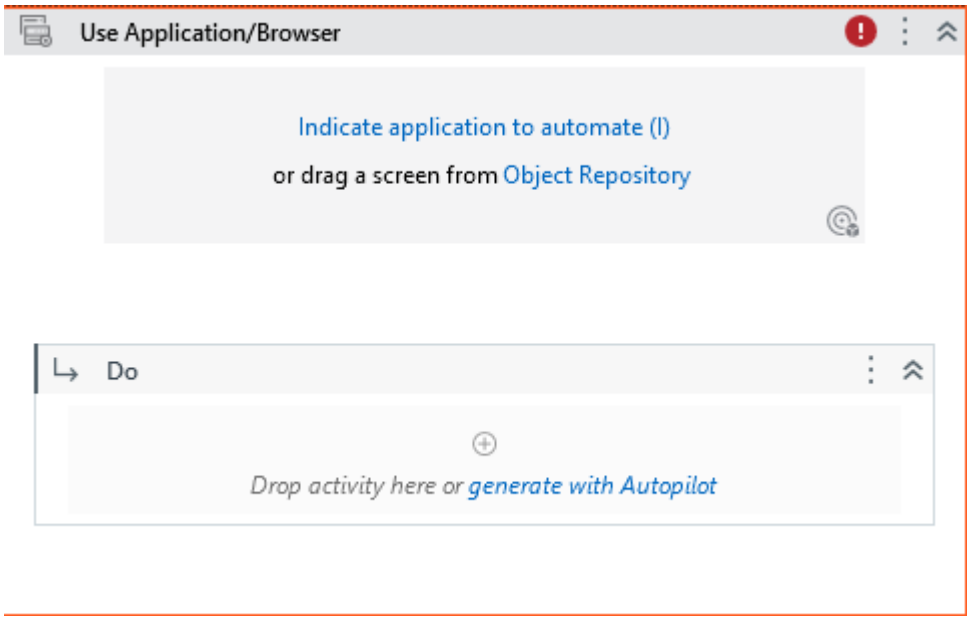


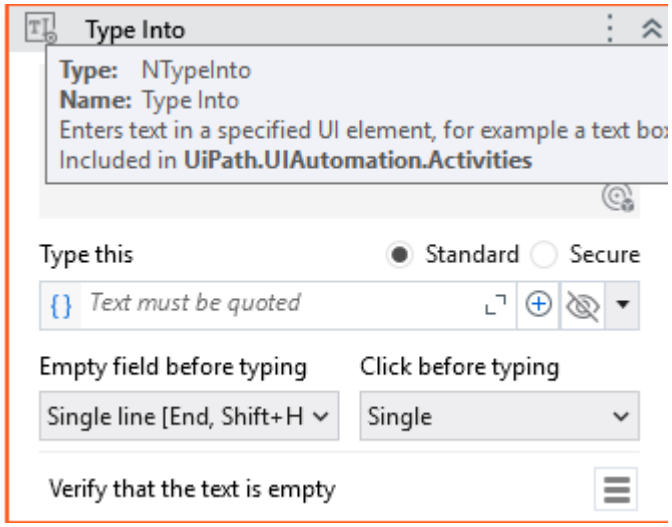
Fig 4.7 Class Diagram

5. MODULE DESCRIPTION

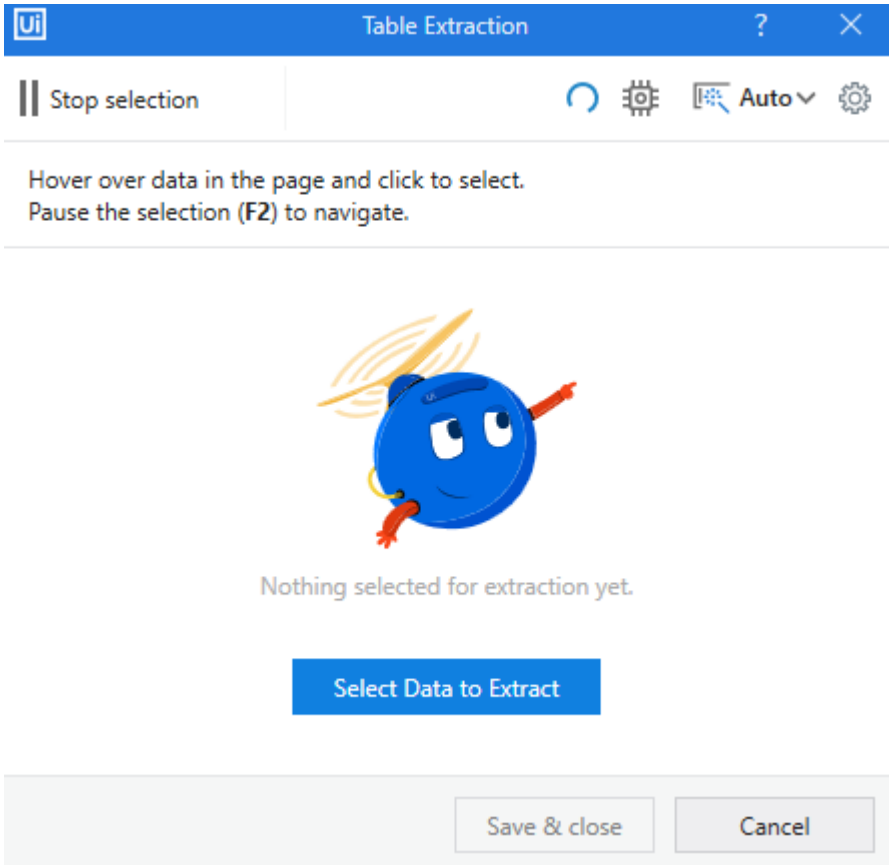
1 Launch Browser - UiPath uses the **Open Browser** activity to launch the e-commerce website and navigate to the product pages you want to scrape.



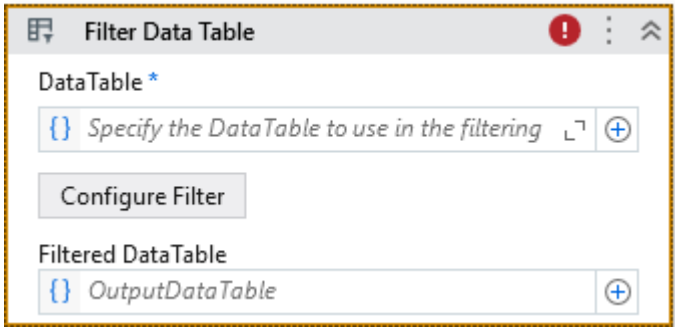
2 Type Into – Used to Scrap the Search Bar for the Product which we want from the User



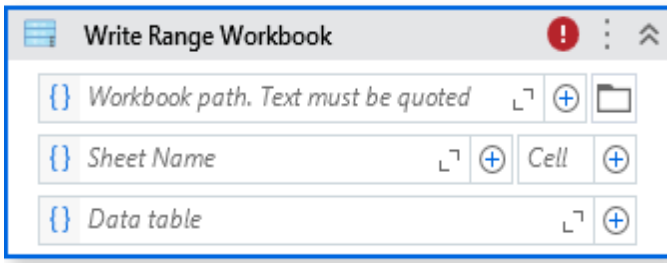
3 Table Extraction - Table extraction in UiPath efficiently captures structured data from web pages or documents, preserving rows and columns for easy processing. It automates the collection of tabular data, such as product details, and stores it in formats like Excel or CSV for further use.



4 Filter Data Table - Filtering a data table in UiPath allows you to extract specific rows based on certain conditions, such as price ranges or product.



5 Write Range Workbook - The **Write Range** activity in UiPath writes data from a DataTable into an Excel worksheet. It allows you to specify the starting cell and the range of cells where the data will be written, either overwriting or appending to existing data.



6. SYSTEM IMPLEMENTATION

6.1 Workflow

1. Open Browser

- **Activity**: Open Browser
- **Purpose**: Launches a web browser (e.g., Chrome, Firefox) and navigates to the target e-commerce website.
- **Details**: In this step, you specify the URL of the website from which you want to scrape the data. You can also configure browser settings (such as maximizing the window or enabling headless mode if required).

2. Extract Data Using Data Scraping

- **Activity**: Table Extraction
- **Purpose**: Extracts structured data from the e-commerce website by identifying tables or repeating elements like product names, prices, descriptions, ratings, etc.
- **Details**: The **Data Scraping Wizard** in UiPath is used to automatically identify the data on the webpage. You click on the first product (or any element you need to extract) and then configure UiPath to capture the entire list or table of product details. The tool will generate a **DataTable** containing all the scraped product information.

3. Filter Data

- **Activity**: Filter Data Table
- **Purpose**: Filters the extracted data based on specified conditions, such as product price, availability, or category.
- **Details**: After data extraction, you can use the **Filter Data Table** activity to keep only relevant rows. For instance, you might want to filter products by price range or remove any out-of-stock items. This activity allows you to specify conditions like "Price > 100" or "Availability = In Stock."

4. Write Data to Excel

- **Activity**: Write Range
- **Purpose**: Writes the filtered and processed data into an Excel sheet.
- **Details**: After processing the data, use the **Write Range** activity to export the **DataTable** into an Excel workbook.

7. CONCLUSION

7.1 Result

The **UiPath-based web scraping bot** automates the extraction of product details such as name, price, URL, offers, and ratings from e-commerce websites. Using UiPath's **Data Scraping Wizard**, the bot efficiently collects structured data while handling pagination, dynamic content, and UI changes. It incorporates **adaptive automation techniques**, such as human-like interactions and strategic delays, to bypass CAPTCHA and anti-scraping mechanisms. The bot can **navigate through multiple product pages**, extract relevant information, and store it in structured formats like Excel or databases. By automating data extraction, it reduces **manual effort**, enhances **accuracy**, and boosts **efficiency** for businesses conducting price monitoring and market research. Built-in **error handling** ensures reliable data extraction, even if website structures change over time. UiPath's automation tools **minimize repetitive tasks**, improving data consistency and decision-making. The bot helps businesses track **real-time pricing trends, competitor strategies, and product availability** effectively. **API integration** can further improve data extraction speed and reduce reliance on traditional scraping methods. Future enhancements, like **AI-driven parsing and ML-based data categorization**, could improve accuracy and adaptability. With **UiPath Orchestrator**, the bot can be scheduled for regular updates, ensuring continuous data collection. Ultimately, the UiPath web scraping bot empowers businesses with **real-time insights**, enabling smarter decision-making in a data-driven market.

7.2 Future Enhancements

1. Advanced Anti-Scraping Bypass Mechanisms – The use of **human-like behavior simulation, headless browsers, and rotating proxies** will improve bot resilience against CAPTCHA and IP blocking.

2. AI & Machine Learning Integration – AI-driven data parsing and ML models can enhance data categorization, detect patterns, and handle unstructured data more effectively.

3. Automated Adaptation to Website Changes – AI-powered scrapers will **self-adjust** to website structure changes, reducing downtime and the need for manual updates.

4. Multi-Platform Scraping – Future scrapers will support data extraction from mobile apps, Progressive Web Apps (PWAs), and voice search-enabled platforms.

8.REFERENCES

1.Mitchell, R. (2018). Web Scraping with Python: Collecting More Data from the Modern Web. O'Reilly Media.

2. UiPath Community Forum. (n.d.). Best Practices for Web Scraping and Automation. Retrieved from https://forum.uipath.com

3. Google Cloud. (n.d.). Cloud Storage for Large-Scale Data Handling. Retrieved from https://cloud.google.com/storage

4. OpenAI. (2024). Advancements in AI-Driven Data Extraction Techniques. Retrieved from https://openai.com/research

5. Selenium Documentation. (n.d.). Headless Browser Automation for Web Scraping. Retrieved from https://www.selenium.dev/documentation/

6.AWS. (n.d.). AWS Data Services for Scalable Storage Solutions. Retrieved from https://aws.amazon.com/data-lakes-and-analytics/

7.Kapoor, A., & Sharma, R. (2021). AI-Powered Web Scraping: Challenges and Solutions. International Journal of Data Science, 34(2), 112-129.

8.Moz. (n.d.). Web Scraping Ethics and Legal Considerations. Retrieved from https://moz.com/blog/web-scraping-ethics

9.Microsoft Azure. (n.d.). AI and Machine Learning for Web Data Extraction. Retrieved from https://azure.microsoft.com/en-us/solutions/ai/