NEWS TEMPLATE ENGINEERING VIA AUTOMATED WEB SCRAPING

|  |  |  |
| --- | --- | --- |
| Piyush Kumar Department of Networking and  Communications, College of Engineering  and Technology, SRM Institute of Science and Technology, Kattankulatur, India [pk0669@srmist.edu.in](mailto:pk0669@srmist.edu.in) | Laraib Hanif Department of Networking and  Communications, College of Engineering  and Technology, SRM Institute of Science and Technology, Kattankulatur, India [lh9435@srmist.edu.in](mailto:lh9435@srmist.edu.in) | L.N.B Srinivas\* Department of Networking and  Communications, College of Engineering  and Technology, SRM Institute of Science and Technology, Kattankulatur, India [srinival@srmist.edu.in](mailto:srinival@srmist.edu.in) |

***Abstract*- This paper is focused on the development of a sophisticated web scraping tool meticulously engineered to extract comprehensive news content from a designated news website. The tool is designed to efficiently retrieve a diverse range of data, including news articles, headlines, author information, publication dates, and any other relevant metadata available on the website Once the data is extracted, it undergoes a seamless transformation process, dynamically populating an HTML canvas template specifically crafted for presenting news content. This template is meticulously designed to ensure optimal readability, visual appeal, and user engagement. The automation of the retrieval and integration process not only saves significant time and effort but also ensures the consistency and accuracy of the curated news layout. By streamlining this process, the project aims to facilitate the generation of visually captivating and structurally organized news presentations that can be easily shared or displayed across various platforms. Moreover, the flexibility of the HTML canvas template allows for customization according to specific preferences or branding requirements, enhancing the overall user experience and brand identity.**

***Terms*—Web Scraping; HTML canvas; Customization; Preferred Templates**

1. INTRODUCTION

This paper presents a dynamic web scraping application meticulously crafted to extract real-time news from various sources using News APIs. Powered by Python's Flask framework and bolstered by BeautifulSoup for parsing, the application efficiently aggregates the latest news content. Users are greeted with a feature-rich interface, providing a diverse array of templates for effortlessly creating downloadable posts. This robust integration of technologies ensures seamless data retrieval and post generation, ultimately enhancing the user experience for accessing and sharing up-to-date news content. Additionally, the application's user-friendly design and customizable features empower users to tailor their news consumption experience according to their preferences, fostering greater engagement and satisfaction. Through its innovative approach, this project seeks to revolutionize the way users interact with and consume news online, offering a convenient and enriching platform for staying informed in today's fast-paced digital landscape.

The paper prioritizes scalability and performance optimization, allowing the application to handle a large volume of requests and deliver news content swiftly to users. Implementation of caching mechanisms and asynchronous processing techniques ensures minimal latency and maximum responsiveness, even

during peak usage periods. In addition to fetching news content, the application incorporates sentiment analysis algorithms to provide users with insights into the emotional tone and sentiment of news articles. By analyzing the sentiment of headlines and articles, users can gain a better understanding of the prevailing attitudes and opinions surrounding particular topics or events.

This paper encourages community engagement and collaboration through open-source development and contributions. By fostering a vibrant community of developers and enthusiasts, the project aims to continually improve and expand the capabilities of the application, ensuring its relevance and usefulness in an ever-evolving digital landscape and emphasizes on security and data privacy measures to safeguard user information and maintain the integrity of the news content. Integration with authentication and authorization mechanisms ensures that only authorized users can access and interact with the application, while encryption techniques protect sensitive data during transmission and storage.

1. LITERATURE REVIEW
2. *Issues and challenges of Web Scraping: Case Study Approach*

In a January 2023 publication in the International Journal of Teaching and Education, a research paper delves into the pivotal role of web scraping in addressing the inherent disarray of internet data. Emphasizing the necessity of extraction and transformation, the paper underscores how web scraping technology has become indispensable in navigating the vast and unstructured landscape of online information. However, the study also sheds light on the challenges that web scrapers encounter in their endeavors. Notably, websites often implement security measures to thwart automated access, posing obstacles to web scraping activities. Additionally, the dynamic nature of websites, characterized by frequent structural changes, presents further hurdles for web scrapers striving to maintain accurate and reliable data extraction processes.

Furthermore, the research paper examines the evolving landscape of web scraping tools and techniques, highlighting the importance of adaptability and innovation in overcoming challenges. It discusses the emergence of sophisticated scraping frameworks and machine learning algorithms designed to navigate complex website structures and evade detection mechanisms. Additionally, the paper addresses the ethical implications of web scraping, including concerns related to data privacy, consent, and responsible use of extracted information.

1. *Web scraping technologies in an API world*

In an article published in April 2013 in BRIEFINGS IN BIOINFORMATICS with the DOI 10.1093/bib/bbt026I and ID 15.5. 788-797, the practicality and efficacy of scraping frameworks are underscored, highlighting their value in establishing data extraction pipelines tailored to meet specific requirements. Despite their utility, the article acknowledges the challenges encountered by web scraping technologies. Ethical considerations and the potential for data inconsistency arising from website modifications are highlighted as prominent concerns. Additionally, the absence of standardized formats in web content adds complexity to data interpretation and integration processes, further complicating the utilization of web scraping tools in an API-centric environment. Nonetheless, the article emphasizes the importance of addressing these challenges through continued advancements in scraping techniques and the adoption of ethical guidelines to ensure responsible data acquisition practices.

The article also discusses the evolution of web scraping technologies in response to the proliferation of APIs (Application Programming Interfaces) and their impact on data acquisition strategies. It explores how scraping frameworks have adapted to complement API-based approaches, offering flexibility and scalability in data extraction processes. Moreover, the article delves into the role of web scraping in augmenting API data by retrieving supplementary information from sources not covered by existing APIs, thus enriching datasets and enhancing analytical insights. Additionally, it examines the importance of interoperability between web scraping tools and APIs, facilitating seamless integration and data harmonization across heterogeneous sources. Furthermore, the article could delve into case studies or practical examples demonstrating the synergistic use of web scraping and APIs in real-world scenarios, showcasing their complementary roles in data acquisition and analysis. By providing a comprehensive overview of web scraping technologies within an API-centric context, the article can offer valuable insights for researchers and practitioners seeking to leverage diverse data sources effectively in bioinformatics and related fields.

1. *Towards data extraction of dynamic content from js Web applications:*

Published in January 2018 with an ID of 324725919, the paper presents a pioneering Web Data Extraction system designed to tackle the complexities introduced by the dynamic nature of JavaScript-based Web development. This innovative system empowers users to establish extraction rules, facilitating the conversion of semi-structured data into a relational format. However, a notable limitation arises when contemporary JavaScript applications render pages server-side, impeding the universal applicability of extraction methods. The paper outlines future research endeavors aimed at integrating advanced DOM extraction techniques to mitigate this constraint, thereby enhancing the system's versatility and efficacy in capturing dynamic content from modern web applications.

Furthermore, the paper delves into the technical intricacies of dynamic content extraction from JavaScript (JS) web applications. It discusses challenges such as the asynchronous loading of data, dynamic DOM manipulation, and single-page application architectures, which complicate traditional extraction methods. The proposed system addresses these challenges by employing innovative algorithms and heuristics to effectively parse and extract data from JS-rendered pages. Additionally, the paper explores the potential impact of browser compatibility issues and cross-origin resource sharing restrictions on the reliability and performance of the extraction process. Future research directions may involve the development of browser extensions or standalone tools to facilitate seamless integration of the extraction system into users' browsing workflows, enhancing usability and accessibility. Moreover, considerations for scalability and efficiency in handling large datasets and continuous monitoring of web application changes are essential areas for further exploration to ensure the long-term viability and effectiveness of the extraction system. Overall, the paper contributes valuable insights and solutions to the ongoing challenges of extracting dynamic content from modern JS web applications, laying the foundation for future advancements in the field of web data extraction.

1. *Web Scraping: Applications and Scraping Tools*

The paper on "Web Scraping: Applications and Scraping Tools" emphasizes the continued significance of web scraping in sourcing valuable data from the internet for Artificial Intelligence and machine learning applications. Its goal is to broaden understanding and expertise by offering insights into relevant tools and practical applications. However, notable deficiencies exist within the paper, particularly in its failure to adequately address the potential legal and ethical implications linked to data privacy and copyright issues. Furthermore, the discussion remains somewhat superficial concerning specific industry challenges and limitations associated with web scraping methodologies. Additionally, while the paper highlights the benefits of web scraping for AI and machine learning, it could benefit from a deeper exploration of the nuances and complexities involved in utilizing scraped data effectively for training models and making informed decisions. Moreover, it could delve into emerging trends and advancements in web scraping technology, such as the integration of natural language processing techniques and deep learning algorithms for more sophisticated data extraction and analysis. By addressing these aspects comprehensively, the paper could provide a more comprehensive and insightful examination of the role of web scraping in the era of AI and machine learning.

Furthermore, the paper could benefit from discussing practical strategies for mitigating legal and ethical risks associated with web scraping, such as obtaining proper consent, respecting website terms of service, and anonymizing or aggregating scraped data to preserve user privacy. Additionally, it could explore case studies or examples illustrating ethical dilemmas and legal challenges faced by organizations utilizing web scraping for AI and machine learning purposes. Moreover, the paper might consider examining the potential implications of regulatory frameworks such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) on web scraping practices, providing guidance on compliance and risk management strategies

Additionally, a more in-depth analysis of industry-specific challenges and use cases could enhance the paper's relevance and applicability to diverse domains, ranging from e-commerce and finance to healthcare and social media.

Furthermore, by addressing emerging trends and innovations in web scraping technology, such as the application of reinforcement learning algorithms for adaptive scraping and the integration of blockchain for data verification and integrity, the paper could offer valuable insights into the future direction of web scraping in the context of AI and machine learning applications.

1. MOTIVATION

**1.Simplifying news access**: This involves creating a streamlined process for users to access news content easily and efficiently. By leveraging web scraping techniques, relevant news articles can be gathered from various sources and presented in a user-friendly interface, eliminating the need for users to navigate multiple websites or applications to find the information they seek.

**2.Showcasing diverse tech tools by merging web scraping with HTML canvas:**

This entails demonstrating the versatility and capabilities of web scraping technology by integrating it with HTML canvas, a powerful tool for creating interactive and visually appealing web content. By combining these technologies, users can experience a seamless fusion of data extraction and presentation, showcasing the innovative possibilities at the intersection of web development and data acquisition.

**3. Making user experience more attractive:**

This involves enhancing the aesthetics and usability of the news access platform through intuitive design, engaging visuals, and interactive features. By incorporating elements such as responsive layouts, dynamic content updates, and personalized customization options, the platform can create a more immersive and enjoyable user experience, encouraging increased engagement and retention.

**4. Adding more value towards educational utility:**

This focuses on leveraging the platform as a valuable educational resource by providing insights into the process of web scraping, HTML canvas manipulation, and data visualization techniques. By offering tutorials, case studies, and interactive learning modules, users can gain practical knowledge and skills applicable to various fields such as web development, data science, and digital media literacy.

1. DESIGN METHODOLOGY
2. **HTML2CANVAS INTEGRATION**

html2canvas is a JavaScript library that allows you to capture screenshots of web pages or specific elements within a web page directly from the user's browser. Instead of relying on server-side rendering or external services, html2canvas operates entirely on the client-side, utilizing the Document Object Model (DOM) and CSS styling to generate an image representation of the webpage as it appears within the browser.

1. **DYNAMIC CONTENT LOADING**

Dynamic content loading refers to the process of fetching and rendering content on a webpage dynamically, often after the initial page load event. Unlike static content, which is predefined and delivered with the initial HTML response from the server, dynamic content is generated or updated dynamically in response to user interactions, server-side events, or client-side scripts.

To extract HTML content, analyze its structure, and retrieve relevant tags using Beautiful Soup, you begin by employing the Requests library in Python to send HTTP requests to the target webpage URL. This allows you to fetch the HTML content of the webpage via a GET request, storing it in a variable for subsequent processing.

1. **TTS MODULE**

TTS stands for Text-to-Speech. It's a module or technology that converts written text into spoken words. In the context of programming and software development, a TTS module typically provides an API or library that allows developers to incorporate text-to-speech functionality into their applications.

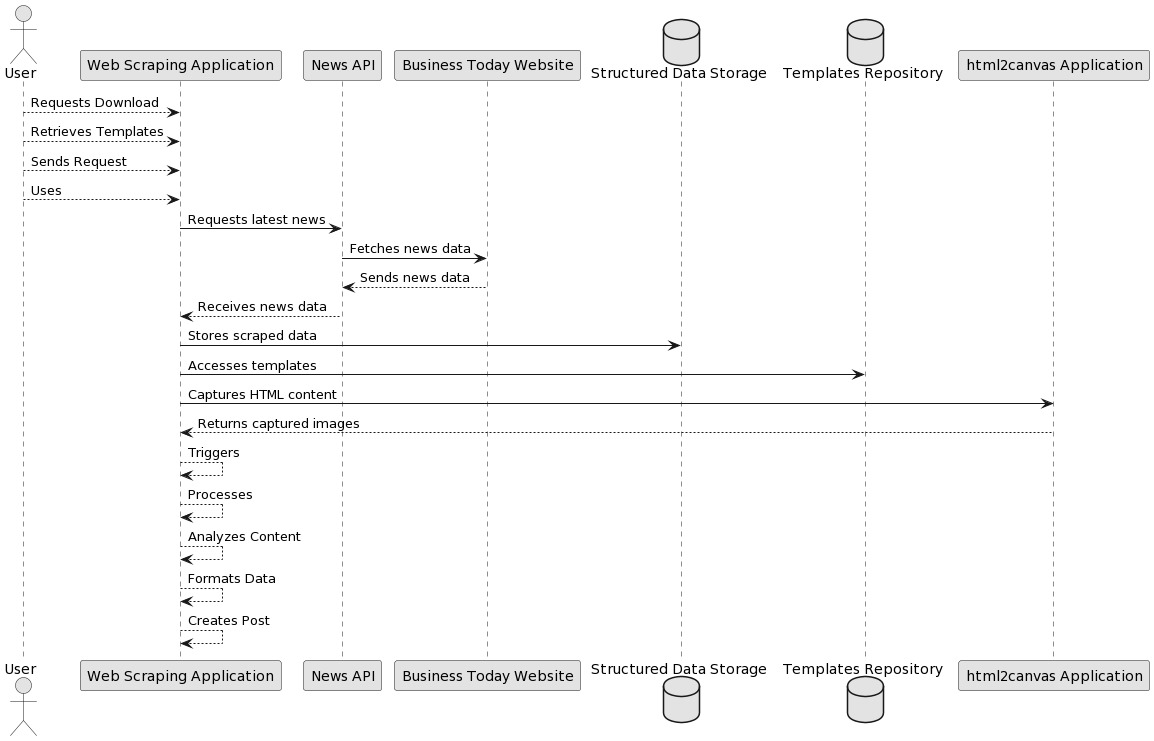
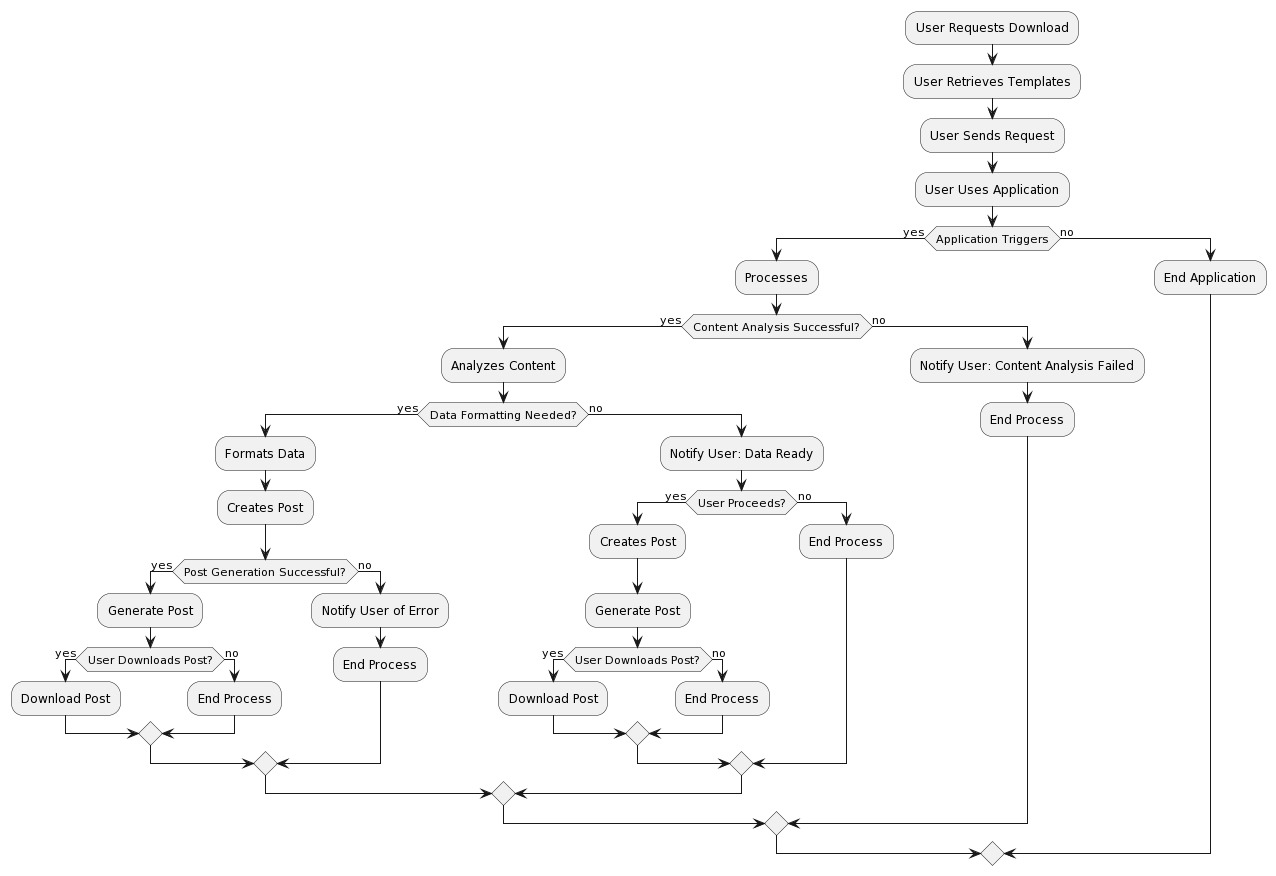
****

Fig. 1. UML Diagram

In (UML) diagrams could be employed to illustrate the system's architecture, including the user interface, scraping and processing components, and post-generation functionalities. Use case diagrams can depict user interactions, while sequence diagrams can detail the flow of data and operations between system components during the scraping process. Class diagrams may represent the structure of the application, including classes for fetching, parsing, and rendering data. Activity diagrams can outline the workflow for scraping and generating posts.The project emphasizes continuous learning and improvement through user feedback mechanisms and data analytics. By soliciting input from users and analyzing usage metrics, the project can identify areas for optimization and enhancement, ensuring that it evolves to meet the evolving needs and preferences of its user base

****

A Use Case Diagram (fig.1) for a Database Management System (DBMS) provides a visual representation of how users and external systems interact with the DBMS highlighting various use cases or functions offered by the system. In the simplified diagram provided, the central element is the "Database Management System" itself.

A flowchart for a Database Management System (DBMS) (fig.2) showing various processes and interactions pertaining to databases. The flowchart demonstrates a systematic diagrammatic picture of the workflow system through major actions and decision points. The “Database System” is the central element of this simplified flowchart where we begin at a “Start” point and end at an “End,” with the user interacting with the DBMS performing specific use cases depicted as distinct branches from his or her interactions.

1. IMPLEMENTATION

Fig. 2. .Flow Chart Diagram

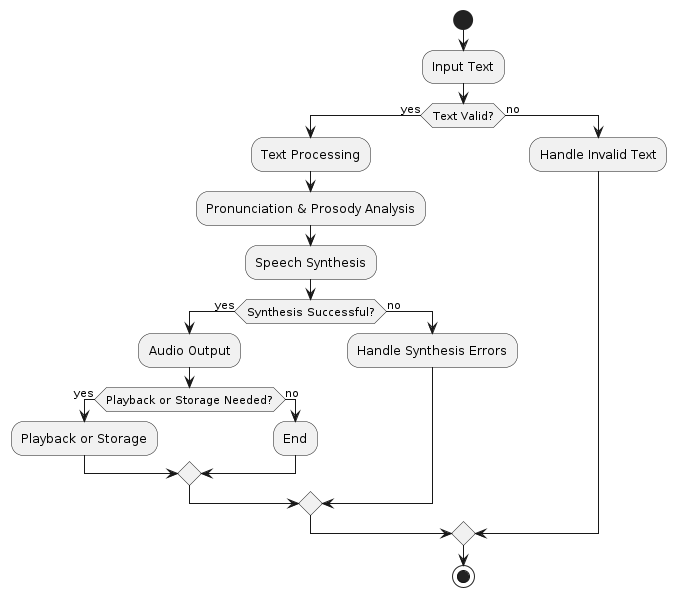


Fig. 3. TTS(Text-To-Speech) Module

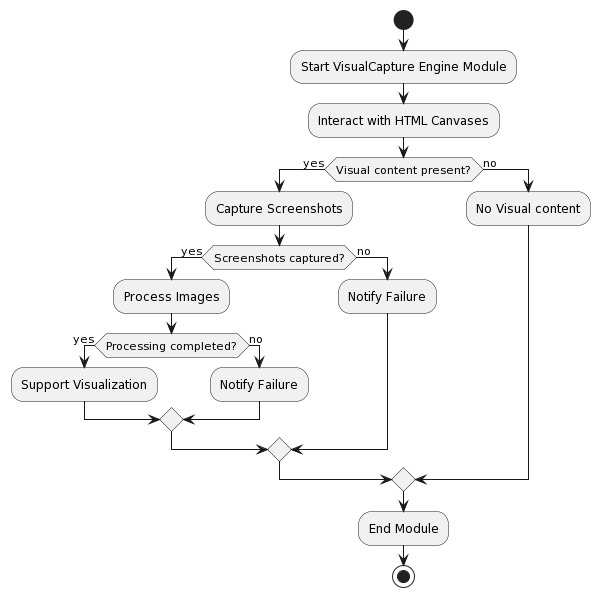


Fig. 4. VisualCapture Module

1. *Modules*

Data Extraction: In the code implementation for data extraction from a news website, we'll focus on retrieving relevant information from the website's pages programmatically. This process involves utilizing web scraping techniques to navigate through the website's HTML structure and extract specific data points such as headlines, article content, publication dates, and author names. Coders employ libraries like BeautifulSoup in Python or Scrapy to parse the HTML content efficiently and extract desired data elements.

Scrapped Data Rendering: The scraped data refers to information retrieved from various online sources using web scraping techniques. This data is typically extracted from web pages and then processed for analysis or presentation in applications. It can include text, images, tables, or other structured content depending on the nature of the website being scraped.

Audio Rendering: Audio rendering using text-to-speech (TTS) technology refers to the process of converting written text into spoken words. This process involves using software algorithms to interpret the text and generate corresponding audio output that mimics human speech. TTS systems can vary in complexity, but they generally involve analyzing the text's linguistic properties, including pronunciation, intonation, and emphasis, to produce natural-sounding speech. The generated audio can be saved as files in various formats or streamed directly for immediate playback to users. TTS technology finds applications in various fields, including accessibility, assistive technology, language learning, and entertainment.

.

1. *Implementation*

In the code implementation, we will delve into how the system operates and executes its functions. This phase encapsulates the translation of algorithms and designs into executable code, bringing theoretical concepts to practical fruition. Here, programmers meticulously craft instructions in a programming language, transforming abstract ideas into tangible actions. This process involves a series of steps, including coding, debugging, and testing, to ensure the reliability and functionality of the system.

Through meticulous attention to detail, coders establish the backbone of software systems, laying the groundwork for their functionality and performance. Moreover, the implementation phase is where creativity intersects with technical prowess, as programmers devise innovative solutions to address various challenges. Ultimately, this stage is pivotal in bridging the gap between concept and reality, as it transforms blueprints into fully functional software systems, ready to fulfill their intended purposes in the digital landscape.

In the code implementation of the login page, we will explore how the system facilitates user authentication and access control. This critical component of software applications serves as the gateway for users to securely interact with the system. Through the implementation of login functionality, users can provide their credentials, such as usernames and passwords, to gain access to protected resources. Coders integrate authentication mechanisms, such as encryption algorithms and secure protocols, to safeguard user information and prevent unauthorized access. Additionally, the implementation encompasses features like password hashing, session management, and multi-factor authentication to enhance security measures. Through meticulous coding practices and rigorous testing, developers ensure the reliability and resilience of the login page, mitigating potential vulnerabilities and threats. Furthermore, user experience considerations are paramount, with attention given to intuitive design elements and error handling to streamline the login process. Ultimately, the implementation of the login page plays a pivotal role in fortifying system security and enabling seamless user interaction.

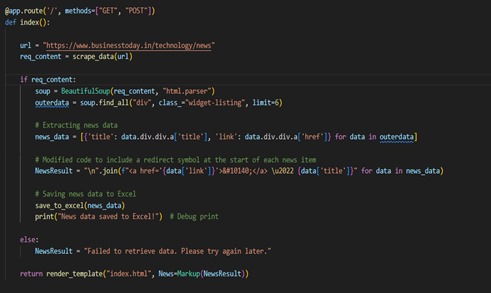


Fig. 5. Data Scraping

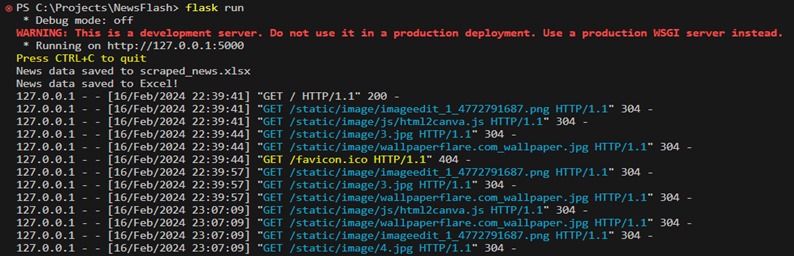


Fig. 6. Template Rendering

1. EXPERIMENTAL RESULTS
2. *Scraped Data Collection and Analysis*

Scraped data collection involves retrieving news articles from a designated website using libraries like requests and BeautifulSoup. The process includes parsing HTML content, identifying relevant tags, and storing data systematically. Analysis encompasses data validation, cleaning, and preprocessing, ensuring accuracy and compliance while iteratively refining the scraping process for optimal performance and reliability.

Match, providing authorization though failed authentication results in denied access. This security approach avoids storing plain text passwords, an additional layer of security. making it difficult for attackers to compromise user. credentials. it challenging for attackers to compromise user credentials.

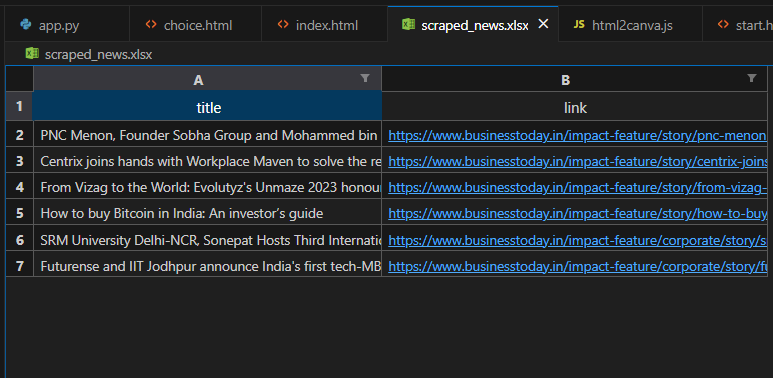


Fig. 7. .Storage collection of scraped data

1. *User News Choice Selection*

This HTML page offers users a selection of categories like tech, stock, business, market, and politics. Clicking on a category redirects users to a news page with relevant content, enabling seamless access to news articles based on their preferences. (shown in fig.8)



Fig. 8. User Choice Selection

1. CONCLUSION

The project “News Template Engineering via Automated Web Scraping” concludes with a robust system for scraping news articles, enriching them with additional features like image downloads and text-to-speech functionality. It ensures accurate data extraction and real-time analysis capabilities, aiming to enhance the visibility of information. With a user-friendly interface, this project significantly improves the accessibility and usability of news content, catering to diverse user preferences and needs.

In summary, the project achieves its objectives by developing a comprehensive web scraping solution tailored for news content with a focus on accuracy and real-time analysis, this project contributes to enriching the dissemination of information in the digital landscape.

REFERENCES

1. K. Sundaramoorthy, R. Durga and S. Nagadarshini, "NewsOne — An Aggregation System for News Using Web Scraping Method," 2017 International Conference on Technical Advancements in Computers and Communications (ICTACC), 2017, pp. 136-140, doi: 10.1109/ICTACC.2017.43
2. E. N. SARR, O. SALL and A. DIALLO, "FactExtract: Automatic Collection and Aggregation of Articles and Journalistic Factual Claims from Online Newspaper," 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS), 2018, pp. 336-341, doi: 10.1109/SNAMS.2018.8554421
3. M. Mohirta, A. S. Cernian, D. Carstoiu, A. M. Vladu, A. Olteanu and V. Sgarciu, "A semantic Web basedscientific news aggregator," 2011 6th IEEE International Symposium on Applied ComputationalIntelligence and Informatics (SACI), 2011, pp. 285-289, doi: 10.1109/SACI.2011.5873015.
4. K. Weiying, D. N. Pham, N. C. Hai and H. H. Ong, "Topic Modelling for Malay News Aggregator," 2018Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA),2018, pp. 1-6, doi: 10.1109/ICACCAF.2018.8776737.
5. R. E. Radu, O. Grigorescu and R. V. Rughiniş, "Security News Aggregator," 2019 18th RoEduNetConference: Networking in Education and Research (RoEduNet), 2019, pp. 1-8, doi:10.1109/ROEDUNET.2019.8909609.
6. Watson, Mark. Scripting Intelligence: Web 3.0 information gathering and processing. Apress, (2009).
7. Le, Quang Thai, and Davar Pishva. "Application of Web Scraping and Google API service to optimize convenience stores' distribution." In 2015 17th International Conference on Advanced Communication Technology (ICACT), pp. 478-482. IEEE, 2015.
8. Mitchell, Ryan. “Web scraping with Python: Collecting more data from the modern web. " O'Reilly Media, Inc., 2018.
9. Barbella, Marcello, and Genoveffa Tortora. “A Semi-Automatic Data Integration Process of Heterogeneous Databases.” Pattern Recognition Letters, vol. 166, Feb. 2023, pp. 134–42.
10. Gupta, M. K., & Badal, D. (2012). Comparative study of indexing techniques in DBMS, available as a conference paper with DOI: 10.5120/333844844.
11. Hsu, Fu-Hau, et al. “TRAP: A Three-Way Handshake Server for TCP Connection Establishment.” Applied Sciences, vol. 6, no. 11, Nov. 2016, p. 358. [www.mdpi.com,](http://www.mdpi.com/) https://doi.org/10.3390/app6110358.
12. Kossmann, J., Papenbrock, T., & Naumann, F. (2021). Data dependencies for query optimization: a survey. The VLDB Journal, 30(5), 761-789.
13. Al-hazaimeh, Obaida M., et al. “Analytical Approach for Data Encryption Standard Algorithm.” International Journal of Interactive Mobile Technologies (iJIM), vol. 17, no. 14, Aug. 2023, pp. 126–43.
14. Eken, Gorkem, et al. “A Lessons Learned Database Structure for Construction Companies.” Procedia Engineering, vol. 123, Jan. 2015, pp. 135–44.
15. D. Kurniawati and D. Triawan, "Increased information retrieval capabilities on e-commerce websites using scraping techniques," 2017 International Conference on Sustainable Information Engineering and Technology (SIET), Malang, Indonesia, 2017, pp. 226-229, doi: 10.1109/SIET.2017.8304139.