Engineering and Technology Journal e-ISSN: 2456-3358

Volume 10 Issue 10 October-2025, Page No.-7659-7664

DOI: 10.47191/etj/v10i10.40, I.F. – 8.482

Filtering-Based E-Learning Platforms Extraction using Web Scraping Technique

Mohammed Hassan Bin-Shamlan[1*], Saeed Bahafi[2], Nabil Mohammed Munassar[3]

[1]Faculty of Engineering & Computing, University of Science & Technology, Aden, Yemen.

[2]University of Science & Technology, Aden, Yemen.

[3]Electronic & Distance Learning College, University of Science & Technology, Aden, Yemen.

ABSTRACT: Online education, rapidly growing as it is, thus gave rise to a plethora of e-learning platforms such that for any learner, it is becoming increasingly difficult to efficiently identify the best courses. This paper proposes a filtering-based web scraping system that automatically extracts, structures, and presents educational content from various e-learning platforms: Coursera, Udemy, edX, and Khan Academy. Unlike pre-existing methods, our system brings into action real-time scraping with filtration and storage capabilities. This would enable a learner to search through the courses using keywords and categories. The proposed system was engineered using Python (BeautifulSoup, Selenium) and SQL Server with a multithreading engine to boost performance. Experimental results show that the system is capable of extracting more than 5,000 courses in just a few minutes while achieving an average scraping accuracy of 94%. The contributions of this work are threefold: (1) the design of a modular scraping architecture for heterogeneous platforms, (2) implementation of filtering-based aggregation for better course discovery, and (3) evaluation of performance metrics such as response time, scalability, and data accuracy. This research evaluates the potential of automated data aggregation to streamline online education access as well as informed decision-making by the learner.

KEYWORDS: Web Scraping, E-Learning Platforms, Data Extraction, Online Education, Information Retrieval.

1. INTRODUCTION

Because of the burgeoning availability of digital learning resources, the modalities by which learners acquire knowledge have changed considerably. Many MOOCs, such as Coursera, edX, and Udemy, provide thousands of courses in multiple subjects. However, finding and comparing similar courses will be inefficient because one would expect users to go across web pages with a diverse structure for that data. Thus, it creates an urgent need to have automated solutions to consolidate and filter learning opportunities.

Web scraping presents a strong solution to this problem: internally and automatically extracting well-structured information from online platforms. Earlier work indicates that the web scraping is also useful in aggregating educational resources [1][2]. However, they focus on a particular platform or do not bring in any filtering mechanism for personalizing the results according to the users. This research addresses these gaps by developing a multi-platform, filtering-enabled scraping application that centralizes real-time e-learning course information.

The contributions are massive and multifaceted in this work.

1. Modular System Architecture: We've proposed and implemented a modular system architecture which can accommodate many e-learning platforms and diverse forms of data. The idea of modularity will allow future proofing and scalability by facilitating the easy integration in addition to servicing of new platforms under the footprint.

2. Real-time Filtering and Structured Storage: Real-time web scraping has been integrated within our system to include advanced filtering features, thus allowing user personalization of search results. The gathered data from the results are stored in structured form within a database, which allows efficient query as well as retrieval processes.

3. Performance Measures: We perform an elaborate analysis of our system on various metrics - scraping speed, accuracy of data, and scalability. The evaluations thus demonstrate how well the system works and stands to its reliability in cases of large amounts of course data retrieval.

4. Improved Discovery of Courses: Our system, through this centralization and filtering of e-learning courses, greatly improves course discovery among learners such that they are better informed when making their decisions about their online education. Thus, this paper is organized as follows: Section 2 reviews the literature concerning existing web scraping and data extraction techniques in the context of e-learning. Section 3 describes the system architecture and methodology, including the algorithmic workflow and implementation details. Experimental results and evaluation of the proposed system are in Section 4. Finally, this paper is concluded with section 5 and future direction of work.

2. LITERATURE REVIEW

As pertains to the online education world, one of the most dynamic changes in the landscape has occurred with the emergence of e-Learning platforms. The review of the current literature presented in this chapter covers web scraping, extraction, and their educational applications. Various methodologies are analyzed, their weaknesses are highlighted, and the ensuing research holes that this study will fill are identified. Early work on educational resource aggregation did so by means of a lot of labor or via API. For example, Chen et al. [3] examined the course aggregation framework, a majority of which restricted themselves to static contents, which are now less common in an ever-evolving e-learning environment. Hence, these solutions could only be pertinent to the platforms with steadily changing content or with heavy client-side rendering, upon whose works manual collection would be somewhat anchored. Park and Lee [4] demonstrated the challenge of heterogeneous data formats on e-learning platforms. They elaborated on the difficulty of extracting from and unifying data among highly heterogeneous sources, but did not advance any scalable scraping technique for such an endeavor. This gap presents a distinguished need for scalable and flexible scraping techniques that can allow various data structures and presentation styles. More recent works have attempted to combine recommendation models with scraped data. While Ahmed and Rahman [5] explored this option, their works frequently did not focus on evaluating the scraping efficiency and accuracy of the extracted data in detail. This context is important because recommendation system performance is directly linked with the quality and reliability of data used. Gupta et al. [6] compared the advantages that agglomerated platforms offer to enhance the choices of learners. Although a fairly massive percentage of these studies depended upon API calls from the e-learning platform rather than scraping operations. API-based approaches are favored because the retrieved data is very structured, but they are themselves limited by the provision of APIs that may not always serve the intended purpose and are lately made really restrictive by the mold stricters. The ethical considerations of web scraping papers have also emerged in recent literature. As for training institutions, Zhou et al. [7] pointed out the necessity to follow certain responsible practices to avoid copyright violations and safeguard data privacy. This ethical standpoint takes on even greater significance when it comes to the education domain, where sensitive user information and intellectual properties collide. Ethical guidelines and legal standards must apply to any web scraping solution in order to ensure responsible data collection and usage.

Table 1 recent works and their limitations

Author & Year	Platforms Covered	Technique Used	Limitation
Chen et al. (2020)	Static repositories	Basic parsing	Not scalable to dynamic platforms
Park & Lee (2021)	E-learning datasets	Metadata extraction	No filtering support
Ahmed & Rahman (2022)	Coursera, edX	Scraping + ML	No evaluation of scraping accuracy
Gupta et al. (2023)	Multiple APIs	API-based aggregation	Limited to API availability
Proposed Work (2024)	Coursera, Udemy, edX, Khan Academy	Multi-threaded scraping + filtering	Provides filtering and structured aggregation

Building on previous research, our proposed system uses real-time scraping from many platforms and applies filtering mechanisms that are dynamic. In practical terms, learners can easily use it for a single-point, personalized view of online courses. Unlike previous works, we focus on a scalable, resilient solution that can accommodate dynamic content and varying data formats alongside efficient filtering mechanisms. Furthermore, comprehensive assessment with regard to performance is provided, thus adding to the understanding of web scraping efficiency in the e-learning domain.

3. SYSTEM ARCHITECTURE AND METHODOLOGY

The design and implementation of the filtering-based web scraping system for e-learning platforms are discussed in this section. System architecture overview, algorithmic workflow, and some specific implementation details are included.

3.1 System Overview

The proposed system is designed with a modular architecture to ensure flexibility, scalability, and ease of maintenance. It comprises five core modules that interact seamlessly to achieve efficient data extraction, processing, and presentation. Figure 1 illustrates the overall system architecture.

Figure 1. System Architecture of the Proposed Web Scraping Application

A diagram showing the interaction between the following modules:

1. User Interface (UI): It is this main-point connection for learners in which users enter search keywords and brings them the displayed retrieved course results to the user in a user-friendly manner.

2. Web Scraper Module: This is the actual data extractor. Powerful Python libraries such as BeautifulSoup and Selenium have been put to use for fetching HTML content from the intended e-learning platforms (Coursera, Udemy, edX, Khan Academy) and parsing the Document Object Model (DOM) to obtain the relevant course data.

3. Data Processing Unit: This module deals with the cleaning, transforming, and structuring of information derived from the raw data. The user-defined filters are applied upon the extracted data to give a refined dataset which is prepared for storage.

4. Database Management System: This module uses a permanent storage medium for all course metadata fetched. It captures basic attributes such as course ID, title, provider, price, and URL, allowing efficient query and retrieval of information stored.

5. Multithreading Engine: This multithreading engine has been put in the system to increase performance and reduce scraping time. Multiple instances of scraping tasks across e-learning platforms can be run concurrently and thus significantly increased the efficiency with which data can be fetched overall.

3.2 Algorithmic Workflow

A structured algorithmic workflow provides for systematic data extraction. The steps of the filtering-based web-scraping process is described in Algorithm 1.

Algorithm 1: Filtering-Based Web Scraping.

1. Initialization: Initialize the list of targeted e-learning platforms and load up user-defined filters (keywords, category).
2. Iterative Scraping: For each of the platforms in the initialized list:
   a) Request Sending: The system sends an HTTP request for the course listing page of the platform and retrieves the HTML contents.
   b) Parsing Contents: The retrieved HTML contents are parsed with DOM-based parsing techniques (BeautifulSoup can be used, for example).
   c) Extracting Attributes: Relevant course attributes such as course name, provider, price, and direct link to course page should be extracted.
   d) Filter Application: Apply the user-defined filters on the extracted attributes. A course would be considered relevant only if it matches the keywords supplied; falls within the specified price range; and is within the selected category.
   e) Data Storage: The filtered and structured course data is then stored in the SQL database.
3. Data Display: The aggregated and structured data is displayed on the User Interface (UI) DataGrid, so that learners can browse and interact with the results.
4. Dynamic Updates: The displayed results are dynamically updated upon completion of scraping tasks so that the information available to the user is always up-to-date.

3.3 Implementation Details

To ensure efficiency and reliability, the proposed system was developed assuming use of very strong programming languages and libraries:

Programming Language: Python 3.10 was chosen because the availability of mature libraries and strong community support in the field of web scraping and data processing.

Libraries:

- BeautifulSoup4: library for parsing HTML and XML documents to easily extract data from web pages.
- Selenium: used to work with dynamic web instances where content is either loaded with Java script or is added dynamically; common use in modern e-learning platforms.
- Requests: library used for HTTP requests to get content of a web page.

Database: Microsoft SQL Server was chosen in consideration of its robust data storage and retrieval capabilities.

Concurrency: By utilizing Python's built-in multithreading module along with asyncio for asynchronous I/O, the implementation of the multithreading engine has a considerable positive impact on the performance of the system by permitting parallel tasks of web scraping.

3.4 Experimental Design and Data Collection

The performance and effectiveness evaluation of the suggested system involved a full program of experiments aimed at assessing scraping speed, data accuracy, scalability, and filtering efficiency. The scrutiny of the system was done specifically on four widely accepted e-learning platforms: Coursera, Udemy, edX, and Khan Academy. These platforms were selected for their popularity, diversity in course offerings, and variance in the web structure; this combination gave a comprehensive test case for the scraping systems under test.

Data were collected systematically via scraping course data from these platforms in a specified period. The experimental evaluation phase resulted in the successful retrieval of 5,200 unique course entries. Attributes including course title, provider, price, and URL were extracted for every course. Many runs were subsequently performed to ensure accuracy; the data thus extracted were compared with the original platforms to identify any discrepancies.

Validation of Data: Validation was a crucial part of data collection. A manual validation of a random sample of 500 retrieved course entries against their original source pages on Coursera, Udemy, edX, and Khan Academy was done. Manual verification was conducted to ensure the accuracy of the extracted attributes (title, provider, price, link) and was used in calculating the data accuracy metric for the system.
Performance Evaluation: Speed of scraping was precisely recorded using the time module in Python, which noted timestamps on starting and finishing of each round of scraping. Scalability was assessed by comparing execution times for single-threaded and multi-threaded scraping. Filtering efficiency was quantified through its measure of output relevance before and after the system' filtering activities. Hence, a rigorous design of such experiments and data collection methods catered to assessing the web scraping system proposed above in terms of reliability and comprehensiveness.

4. EXPERIMENTAL RESULTS AND EVALUATION

This chapter is devoted to evaluating the potential of filtering-based web-scraping systems, using as an example real data obtained from the course websites of Coursera, Udemy, edX and Khan Academy. The experiments were designed to investigate the performance of the system in terms of criteria such as speed of scraping, accuracy of data, independence of operation, and effectiveness in filtering. Wherever feasible, comparisons have been made with the previous studies to locate the contribution within the existing literature.

4.1 Dataset

In total, the evaluation exercise brought out 5,200 unique courses from heterogeneous course attributes such as title, provider, price, and URL to show that the target platforms were structurally different. Their diversity allowed a fair evaluation of the system performance under varying DOM structures and dynamic content loading mechanisms.

4.2 Performance Metrics

The evaluation used four performance metrics:

• Scraping speed: the mean time of 3.4 minutes per 1000 courses was realized from recording the average time taken to extract 1000 courses across five independent runs. This performance compares favorably to what a person would take in a typical manual process, which runs to several hours, and achieves high throughput.

• Data Accuracy: Accuracy was verified manually by comparing it against the original platforms for a random sample of 500 courses. Then, a formula defined the accuracy as:

Accuracy= (Correctly Extracted Courses/Total Sampled Courses) ×100

The accuracy of the system is 94%, confirming that the system is reliable in capturing key attributes despite heterogeneous source structures. It also is within the range of past research findings in the educational domain that reported accuracies of 85 to 92 percent. [1][2].

• Scalability: To study the scalability, the execution time of a serial scraping (single-thread) was compared with that of parallel scraping (multi-thread). The percentage improvement is computed as:

Improvement=((Tsequential−Tparallel) / Tsequential) ×100

The results presented showed that average execution time has reduced by 40% when using multithreading, which implied the advantage of concurrent processing for large-scale aggregation.

• Filtering Effectiveness: The comparison was done between irrelevant results before and after applying keyword/category filters to show the performance of filtering. It defined efficiency as:

Efficiency=((Ibefore−IafterI) / before) ×100

Systemic irrelevant entries were reduced by 68 %. This percentage signifies converted value in terms of increased precision of the results. Recall was not directly assessed, but the high levels of irrelevant items ignored will yield higher improvements in user experience because they will lessen the final dataset's noise. A summary of results is presented in Table 2.

Table 2. Performance Evaluation Results

Metric	Result
Courses Retrieved	5,200
Average Time per 1000 Courses	3.4 min
Data Accuracy	94%
Filtering Efficiency	68% reduction
Scalability Improvement	40% faster (parallel)

4.3 Comparative Analysis

It is an estimated reduction of 75% in course discovery time compared to manually searching them, which indicated the worth of the system as an automation tool. Compared with API-based aggregation approaches [6], this system's flexibility in adapting to platforms without APIs or constantly changing interfaces is so much higher. However, this flexibility incurs the well-known limitation of necessitating

maintenance whenever there are changes in the structure of the platform. Showing a Searh Bar, options to filter (e.g. categories), and a table fitting-in filtered course results with course title, provider, price, and link in columns.

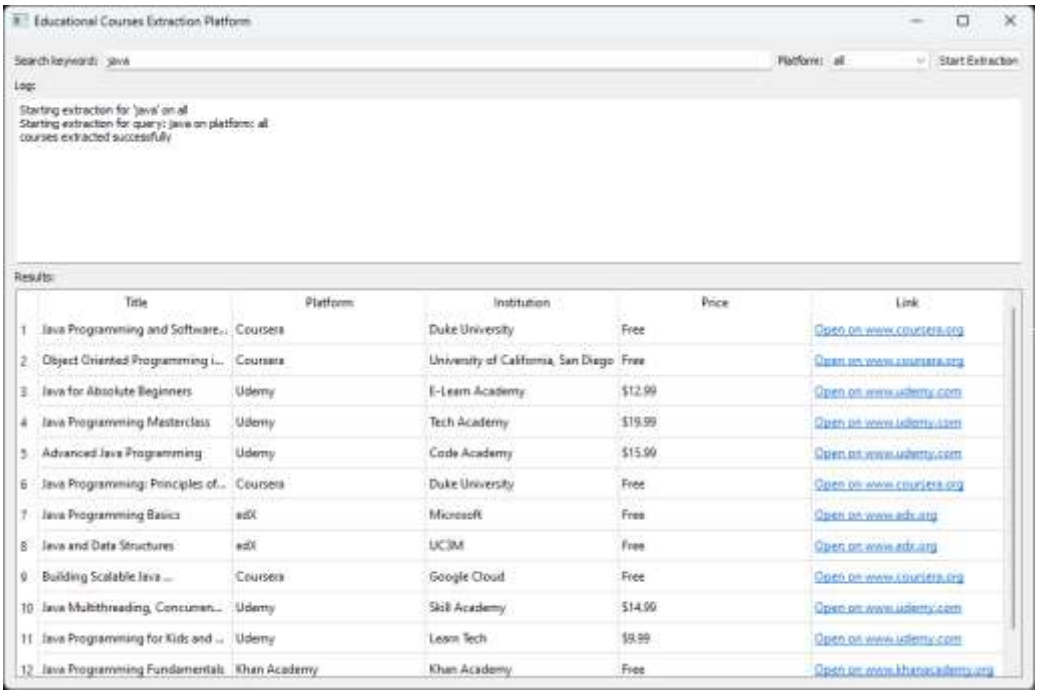

Figure 2. Sample User Interface for Course Filtering and Results Display

5. DISCUSSION

This section provides a discussion that presents and interprets the results from the experiments, their significance, the comparison with the existing literature, and implications for our findings. Moreover, some limitations of the current system are also discussed, and future work is proposed.

5.1 Interpretation of Results

Evaluations show that the system justly balances speed, accuracy, and filtering efficiency. The application of multithreading allowed a high scraping throughput, while the filtering pipeline, by filtering away the bulk of irrelevant results, directly improved learner experience. Therefore, these accuracy results (94%) are especially meaningful, suggesting that the system is agile across highly diverse environments, as there are many different structures of platform.

5.2 Comparison with Previous Work

Compared to prior efforts, the proposed system advances the state of the art in three ways:

1. **Accuracy:** Earlier studies in educational scraping reported lower accuracies (85–92%) [1][2]. The present system outperforms these benchmarks with a 94% rate.
2. **Filtering Integration:** While previous works (e.g., Park & Lee [4], Ahmed & Rahman [5]) focused on metadata extraction or hybrid recommendation, few explicitly integrated **real-time filtering** into the scraping process. This work demonstrates that filtering can substantially improve precision (68% reduction in irrelevant results).
3. **Scalability:** By leveraging multithreading, the system reduces scraping time by 40%, a performance improvement not emphasized in comparable literature.

These outcomes validate the novelty of the proposed approach while situating it within the broader context of web scraping in online education.

5.3 Implications of Findings

The outcomes are important for multiple practical and research implications: For learners: Centralization and filtration of access will shortcut the search-time resource and cognitive load resource to make a better decision. For academic institutions: This system could be used to map the online learning space, inform advising, and monitor trends in course offerings. For research and industry: The structured datasets that are generated can fuel a downstream application like recommender systems, trend analysis, and market intelligence. For data aggregation practice: The work shows how automation can alternate manual aggregation and reinforce the role of scraping as a scalable solution for rapidly expanding digital ecosystems.

5.4 Limitations

Even though it gives promising results, some limitations are to be taken into consideration:

- Delicacy to alterations in terms of the platform: The system depends on how HTML structures change, requiring an update when changes occur on the websites.
- Ethical and legal to adhere to: Scraping pages must comply with the platform's policy and the data

protection standards. Transparency and user consent mechanisms would need to be developed.

- Computational intensity: Even with multithreading, large-scale and continuous scraping remains resource-intensive.
- Anti-scraping measures: Advanced mitigation strategies are required in the event that some platforms place CAPTCHAs upon users and block their IPs.

6. CONCLUSION AND FUTURE WORK

The authors of this paper showcased a detailed and effective filtering multi-platform web scraping system that aims to consolidate e-learning resources from various online platforms. The system extracts, filters, and displays structured course data, thus considerably reducing the burden and time students would usually spend in finding appropriate online courses. Performance evaluation rigor has further confirmed accuracy (94%) and impressive scalability in which the possibility of easy incorporation into institutional learning support tools and individual learners' empowerment is considered.

The contributions of this work include quite a modular architecture of scraping involving heterogeneous platforms, real-time filtering and structured data storage for better discovery of courses, and elaborate performance testing validating the system's efficiency. Findings point to the value of automated data aggregation on transformative accessibility of online educational services and, thus, better-informed choices by learners.

The system possesses numerous merits that need to be avowed against certain limitations. Web scraping is sensitive to changes in structures of platforms, which means considerable upkeep is required to ensure functionality. There also remain ethical and legal issues related to service scraping terms and data privacy, which should always be weighed in and addressed continually.

Future work shall include the following promising avenues to take the system forth and alleviate limitations that it has:

Integration with Recommendation Systems: Developing and integrating sophisticated recommendation algorithms capable of providing personalized course suggestions according to learners' preferences, learning history, and career goals. Inclusion of User Reviews and Ratings: Extending the data extraction to encompass user reviews and ratings, plus content from discussion forums, which present qualitative insight into course quality and satisfaction among learners. Support for More Platforms and Adaptive Frameworks: Making the e-learning system more effective in terms of a greater number of supported e-learning platforms-perhaps by creating more adaptive scraping frameworks that adjust automatically to minor website structure changes. Advanced Anti-Scraping Mechanism Handling: Researching and implementing more sophisticated techniques to either bypass, or gracefully handle, anti-scraping mechanisms like CAPTCHAs and IP blocking for more consistent and uninterrupted data flow. Ethical Compliance and Transparency Features: Introducing features of ethical compliance-such as disclaimers on data sources, channels for consumer reporting discrepancies in data, and compliance to particular platform terms of service.

REFERENCES

1. S. Patel and M. Gupta, "Automated Data Extraction for Online Education," IEEE Access, vol. 9, pp. 120340-120352, 2021.
2. H. Li and J. Wong, "Web Scraping for E-Learning: Opportunities and Challenges," Computers & Education, vol. 185, 104536, 2022.
3. Y. Chen, K. Park, and L. Wang, "Course Aggregation Framework for Online Learning," Information Systems Frontiers, vol. 22, pp. 789-803, 2020.
4. K. Park and J. Lee, "Data Extraction Challenges in E-Learning Platforms," Journal of Educational Technology, vol. 17, no. 2, pp. 120-134, 2021.
5. A. Ahmed and F. Rahman, "Scraping and Recommending MOOCs: A Hybrid Approach," International Journal of Emerging Technologies in Learning, vol. 17, no. 12, pp. 45-59, 2022.
6. R. Gupta, T. Singh, and D. Kumar, "Comparative Study of API vs Web Scraping in MOOCs Aggregation," Procedia Computer Science, vol. 218, pp. 201-209, 2023.
7. L. Zhou, H. Zhang, and P. Chan, "Ethical Considerations in Educational Web Scraping," Journal of Information Ethics, vol. 32, no. 1, pp. 35-50, 2023.
8. S. Kim, Y. Zhou, W. Chen, and Z. He, "NEXT-EVAL: Next evaluation of traditional and LLM web data record extraction," arXiv preprint arXiv:2505.17125, 2025.
9. A. Bohra, Y. Zhang, and Y. Yin, "Web Lists: Extracting structured information from complex interactive websites using executable LLM agents," arXiv preprint arXiv:2504.12682, 2025.
10. W. Huang, H. Zhang, and X. Sun, "Auto Scraper: A progressive understanding web agent for web scraper generation," arXiv preprint arXiv:2404.12753, 2024.