International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 5 May 2023, pp: 94-97 www.ijaem.net ISSN: 2395-5252

University Results Extraction and Analysis Tool using Batch Processing Implementation System

Kedar C Pai, Rohan Nayak K, Sindhu Badal, Suryanarayan U Bhat, Mrs.Geethalaxmi

[1,2,3,4] *Student, Dept of Information Science and Engineering, Canara Engineering College, BenjanapadavuBantwal Taluk, Karnataka*

[5] *Assistant Professor,Dept of Information Science and Engineering, Canara Engineering College, BenjanapadavuBantwal Taluk ,Karnataka*

Date of Submission: 01-05-2023 Date of Acceptance: 08-05-2023

ABSTRACT: This research paper is about implementation of a desktop application for extraction and excel sheet generation of results from a university website. This will be mainly implemented using Web technologies and Python Automation modules.

We will be using Electron JS which is a popular framework for developing desktop applications. On the backend side of the system, we will be using Node JS as an API for linking front end and backend systems. Backend will consist of Python executables files for processing the data & actions.

We will be using batch processing for parallelly executing and processing of the extracted web scraped results data.

KEYWORDS: Automation, Batch Processing, Web Scraping, Python,, JavaScript.

I. INTRODUCTION

The existing procedure requires a specialised coordinator, teacher, or other related authority to manually process the results of university students from a certain batch in a college. Since the university website is updated and the university has added a security layer of captcha and anti-code sniffing to prevent scarping data from their result domain, the existing Web scraping solutions are out of date.

Through a session request, our project extracts the data from a university results website and processes it to create an excel sheet report. This will automate the tiresome process of manually entering data and provide the user the opportunity to perform extra processing by giving necessary additional parameters.

Batch processing is what we'll be doing.

II. IMPLEMENTATION

The most crucial and last phase of software development is implementation, which must be finished. Usage examples include accepting particular specifications or calculations, transforming them into a framework, programme, or product segment with the aid of computer programming, and releasing them into the environment where they are intended to function.

The entire application system has various intermediatory steps like frontend view module, extraction module, batch processing module and database cloud storage modules.

The backend of the application is mainly powered by Node JS, Python and MongoDB. The Python process are rendered as executables and run as child process through the Node JS API. The backend server is inbuild within the desktop application without any service dependency. OpenPyxl, xlwings ,chompjs , json , requests are python modules are used for performing excel, http request-response and json operations

III. DESIGN

The following is a list of the design and implementation constraints: To avoid DDOS attacks from the application on the host server, the USN range for generation should be constrained.

- The server answers for the server refresh period, which is often reported to be between 20 and 25 requests and lasts 30 seconds, must be handled appropriately.

- The end user must receive the appropriate acknowledgment when the server responds with an Invalid USN or an Out-of-Range USN, for example.

- The interface design should also make it simple for the end user to use the available options.
- The system should be capable of coping with network and invalid error constraints, such as network timeout, disconnections, and erroneous captcha conditions.

These three categories of key result server constraints must be addressed.

The system's primary features are USN input format and validation in accordance with university standards. In manual extraction, the user must enter the captcha; in its automatic counterpart, the system will enter the captcha. The system must check and verify the path to the system file and directory. Only a machine with a working IP address and a secure network connection should be able to access the programme. To avoid abuse of the programme, it should be made sure that it cannot be altered, injected, or embedded in any other runtime environment. Only authorised application admins may have access to certain product features.

Here, As referred in FIGURE 1 an authorised user may access the application, fill out the form with the necessary information, and then browse to the desired results. After the form data has been validated, the user can extract the data and create an excel file if the data is accurate. Another choice is to display the excel inside the programme as a table. It will obtain the information from the user-selected file. Here, we can learn how users will engage with the programme and what features they may anticipate from it.

In addition to students, the main actors in this system are the college staff and teachers. Principal Use Case Activities Include:

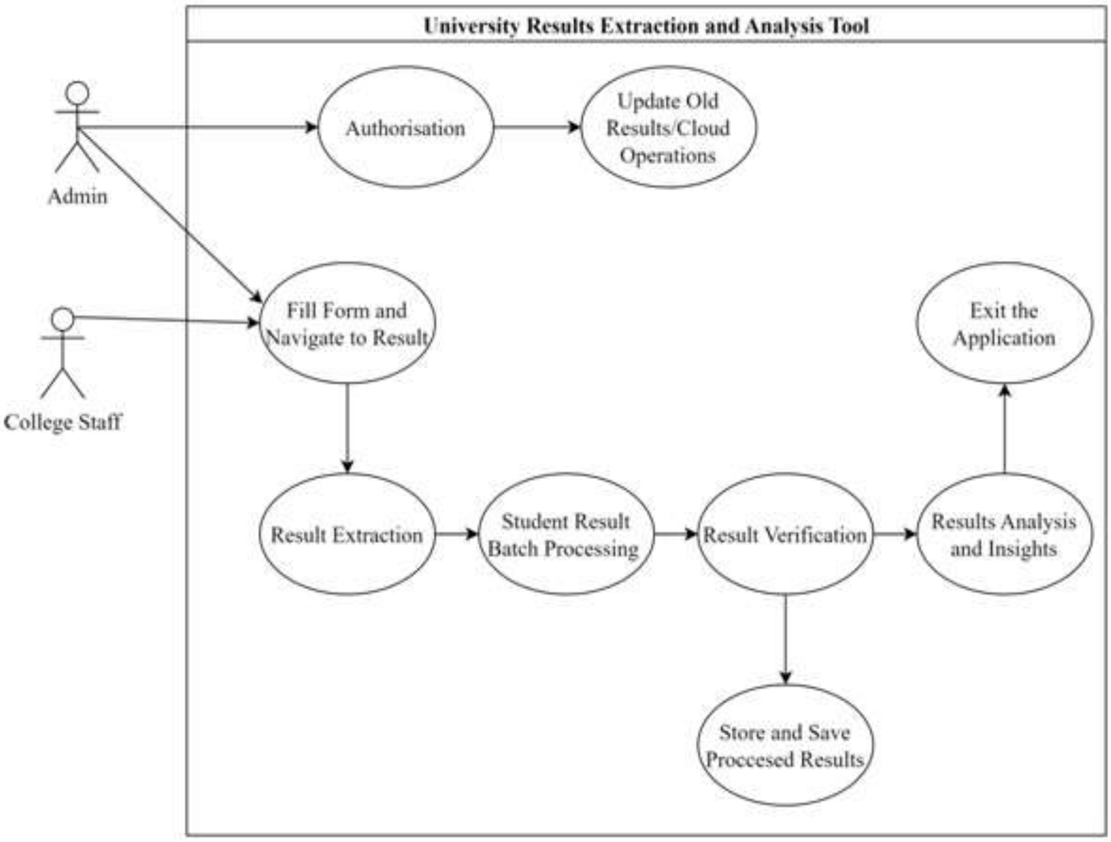
University Results Extraction and Analysis Tool
Admin
College Staff
Authorisation
Update Old Results/Cloud Operations
Fill Form and Navigate to Result
Exit the Application
Result Extraction
Student Result Batch Processing
Result Verification
Results Analysis and Insights
Store and Save Proccesed Results

Figure 1 DESIGN DIAGRAM

IV. METHODOLOGY

The scraping of online results is carried out using standard JavaScript; the markup from the Document Object Model (DOM) is parsed, and the data is then reorganised into the required data structure and forwarded to be processed further. Until we have traversed the entire USN input range, the Captcha and Token are stored because they will be used as a key value in the post request object that will be sent across with the updated USN value.

The analysed results box will now have choices for calculations and sorting, including top performers, average grades, and pass/fail percentage. Every time the server answers and the browser request a page, the Batch Processing of results is running in the background as the browser navigates through each student in the USN range.

The university results page will be opened within the instance and scope of the application. The user can then the user can navigate to results page and fill the form one time with USN and captcha details, after successfully fetching results of first student, the user can select Auto Next option from application menu or context menu, it will pop up a window asking for range of results, the user wants to extract. The user can enter the range details and submit, this will begin the automatic extraction and web scrape the results data and store it in a appropriate data structure , after the given range of results is traversed the application will redirect the user to the generation page , where he can enter the additional input details like branch, semester , academic year and subject credits to calculate SGPA, percentage and other details.

V. RESULT AND EVALUATION

For the college facility, we were able to successfully develop an intuitive desktop application to automate the process of result extraction and generation of excel sheet reports. Building this ecosystem has lowered the whole laborious time work of hours combined to 5–10 minutes of effort.

Recent findings from the university's domain were used to test the application, which produced correct results and effectively generated an excel sheet report without the burdensome and time-consuming manual data entry procedures.

When the server's internal speed and response time are consistent, it takes around 4-5 minutes to generate and process results for 60 students. However, this time may vary depending on network speeds and the results server's request-response timings.

VI. CONCLUSION

The programme has a lot of room to grow in the future, and we may include more types of visualisations. graph modules that display graphs and charts for the visual representation of data that has been retrieved and processed. The end user may produce and store the results in many export formats. Using the native Excel charts API, charts and graphical representations may easily be added to an excel template. The data gathered over time may be utilised as a dataset for machine learning models to be implemented and predictions of department, topic, or student results based on historical performances and data inputs made. When a student's results are updated following the university's release of the revaluation results, bots can also be set up to manually update the results.

VII. FUTURE SCOPE

The programme has a lot of room to grow in the future, and we may include more types of visualisation. graph modules that display graphs and charts for the visual representation of data that has been retrieved and processed. The end user may produce and store the results in many export formats. Using the native Excel charts API, charts and graphical representations may easily be added to an excel template. The data gathered over time may be utilised as a dataset for machine learning models to be implemented and predictions of department, topic, or student results based on historical performances and data inputs made. When a student's results are updated following the university's release of the revaluation results, bots can also be set up to manually update the results..

REFERENCES

[1]. GauripriyaKakati,KomalRevankar, PratidnyaKittur, 2022 “ WEB SCRAPING FOR VTURESULTS ”, International Journal of Current Engineering and Scientific Research (Ijcesr)

[2]. SJohn W. Fowler, Lars Mönch, A survey of scheduling with parallel batch (p-batch) processing, European Journal of Operational Research, Volume 298, Issue 1, 2022,Pages1-24,ISSN03772217, https://doi.org/10.1016/j.ejor.2021.06.012

[3]. R. N, A. M. S and C. N. Babu, "An Optimal Data Entry Method, Using Web Scraping and Text Recognition," 2021 International Conference on Information Technology (ICIT), 2021, pp. 92-97, doi: 10.1109/ICIT52682.2021.9491643

[4]. D. M. Thomas and S. Mathur, "Data Analysis by Web Scraping using Python," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA),2019, pp. 450-454, doi: 10.1109/ICECA.2019.8822022.

[5]. T. Karthikeyan, K. Sekaran, D. Ranjith and J.M. Balajee, "Personalized content extraction and text classification using effective web scraping techniques", International Journal of Web Portals, vol. 11, no. 2, pp. 41-52

[6]. V. Shrivastava, "A Methodical Study of Web Crawler", Journal of Engineering Research and Application, vol. 8, no. 11, pp. 01-08

[7]. Khadka, B., 2019. “Data analysis theory and practice: Case: Python and Excel Tool”

[8]. Gunawan, R., Rahmatulloh, A., Darmawan, I. and Firdaus, F., 2019, March. Comparison of web scraping techniques: regular expression, HTML DOM and Xpath. In 2018 International Conference on Industrial Enterprise and System Engineering (ICoIESE 2018) (pp. 283-287).Atlantis Press

[9]. A. V. Saurkar, K. G. Pathare and S. A. Gode, 2018 "An overview on web scraping techniques and tools", International Journal on Future Revolution in Computer Science &Communication Engineering, vol. 4, no. 4, pp. 363-367

[10]. D. Pratiba, A. M.S., A. Dua, G. K. Shanbag, N. Bhandari and U. Singh, "Web Scraping And Data Acquisition Using Google Scholar," 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), 2018, pp. 277-281, doi: 10.1109/CSITSS.2018.8768