

University Results Extraction and Analysis Tool using Batch Processing

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

¹Assistant Professor, Dept of Information Science and Engineering, Canara Engineering College, Benjanapadavu Bantwal Taluk, Karnataka

^{2,3,4,5}Student, Dept of Information Science and Engineering, Canara Engineering College, Benjanapadavu Bantwal Taluk, Karnataka

Submitted: 18-12-2022

Accepted: 31-12-2022

ABSTRACT: This research paper is about developing a desktop application software to create Excel Sheet Report of the University marks of the students of the college. Every semester the results are declared on university official website, the result is available on the respective result subdomain. But the option to download all the results of a class at once was not possible for the college or an individual looking for the result of the entire class. Only way was to manually prepare an excel sheet by inputting the data of each student.

The current system of preparing excel sheet report of student marks is extremely tedious and time consuming, where in the marks of the student have to manually entered by a person, individually doing the data entry by checking the results of student from the University results website. This project is created to automate the process of excel report preparation manage the results of students. The project is planned to be built using tools and technologies relevant with automation process in the industry like web scraping and the technique used for mass processing like batch processing.

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

I. INTRODUCTION

The current process to batch processes the results of university student of a particular batch in a college has to be done by manual intervention of a dedicated coordinator, faculty or other concerned authority. The existing Web scraping solutions are outdated as the University Website is updated and also University has added a security layer of captcha and anti-codesniffing to prevent scraping data from their result domain.

Our project extracts the data from a university results website through session request and process it to prepare excel sheet report. This is highly useful as it will automate the tedious task of manual data entry and give the user the option to do additional processing by providing necessary additional parameters.

We will be using Batch Processing technique to process the set of data simultaneously for performance. The entire set of data will be processed together without any human intervention.

The previously used methods have become obsolete and in practical to use as the structure of the website has changed and some features are added for security purpose. Therefore, creating a web crawler that suits the current web structure with ability to process and sort the results has become essential. Hence considering the need for automation of this manual process, our project aims to come as aid to solve this problem and provide a foundational tool for result extraction with scope to include more functionalities.

II. PROBLEM STATEMENT

Despite advances in technology, nowadays, University result batch processing for creating excel sheet report is often done manually by an individual. This report preparation and segregation can be automated by python, which is popular for providing module support for many automated tasks.

When performing a long and tedious task, the tendency to do errors increases thus resulting in lower accuracy rate of the task. Wrong data entry can lead to number of problems and unnecessary effort to rectify verify and correct them at later

stages. At this age of fast computation, it can prove beneficial to automate such tasks to save time and

get highest accuracy rate by computer processing methods.

III. ARCHITECTURE AND DESIGN

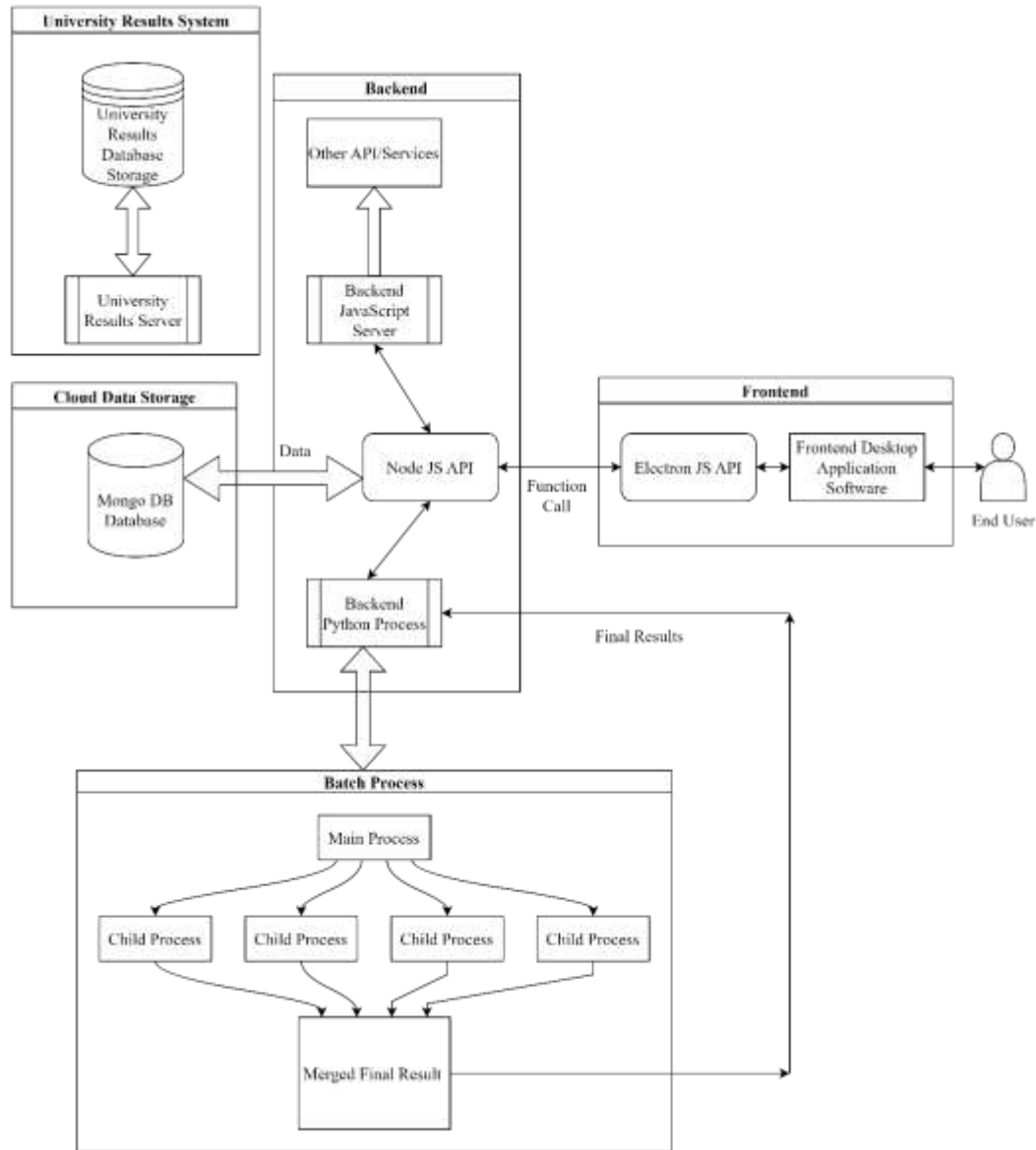


FIGURE 1 ARCHITECTURAL DESIGN

The frontend is implemented with a JavaScript frame work Electron.JS, it is a runtime framework that allows the user to create desktop suite applications with HTML5, CSS, and JavaScript. 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 inbuilt within

the desktop application without any service dependency. OpenPyxl and json python modules are used for performing excel and json operations,

We will be storing the results on cloud drives for backup also for certain fixed type of information like college faculty and roles we will be using a NoSQL database like MongoDB Atlas. Cloud Storage like Google Drive will be used for maintaining excel sheet backup records. The Overall architectural flow and design of

components of the system is depicted in the FIGURE 1 above.

IV. METHODOLOGY

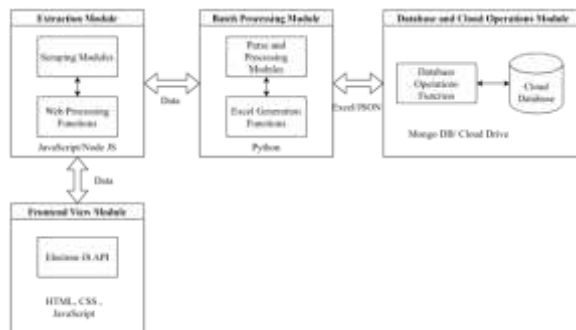


FIGURE 2 MODULAR FLOW DESIGN

The entire application system has various intermediary steps like frontend view module, extraction module, batch processing module and database cloud storage modules, the modular design and flow is represented in the FIGURE 2 above,

The steps for extraction, processing and generation of excel sheet is as follows:

- Step [1]Start
- Step [2] Navigate to the university results page using inbuilt render of website within the software.
- Step [3]Fill the form with correct details.
- Step [4]Captcha and Token is recorded and stored for later stage.
- Step [5]Decision for Auto Traversal or Manual Traversal.
- Step [6]If auto traversal is chosen, the system will automatically traverse all the results and stop the traversal for server refresh for 30 seconds, during this time the users screen is alerted with a timer for 30 seconds indicating the refresh, after 30 seconds the traversal will automatically proceed till the range maximum.
- Step [7] Based on the selected traversal method we will resend the POST Request using the Token and Captcha recorded earlier, The Form object is created with these key and values pairs with updated USN.
- Step [8]Fetch the Results Page Response from the University server.
- Step [9]Parse the HTML.
- Step [10] Process the results using batch processing method.
- Step [11]Generate and Export the results in Excel or JSON format.
- Step [12]Save and Store the Results on the cloud Database.

- Step [13] View and Analyse the results, Export and Save Excel sheet option.
- Step [14]End

The main purpose of implementing timeout of 30 seconds is to prevent server request overflow and exhaust, this break of 30 seconds will prevent the server from DDOS attack.

The Scraping of web results is implemented by using vanilla JavaScript, the markup is parsed of the Document Object Model(DOM) and the data is restructured into the necessary data structure and sent forward to for further processing.

The Captcha and Token are stored as they will be used as a key value in the post request object which will be sent across with the updated USN value until we traverse through the entire USN input range.

The analyse results window will currently have options to do calculations and sorting like calculating pass v/s fail percentage, toppers, average marks etc.

The Batch Processing of results is happening in the background during traversal of each student in the USN range, every time the server responds and the DOM updates with the latest result, the previous parsed data structure is processed. Finally, all the collected data structure is mass processed with to generate the required excel sheet using the python child process executable.

The user is also provided browse path option to select the folder path for excel generation, the ability to view the generated excel sheet within the application will also be provided for easy access and quick verification.

An exclusive form interface will be provided to update the result of the student manually in addition to update using the

reevaluation updating by traversal of results from within the application.

Proper authorisation and authentication of the end user will be carried out through password mechanism for authenticating the user operations, some of the actions like deletion or updating of results can be limited to the admin of the application as configured by the college. Operations like retrieval, analysis and generation of results can be kept public for anyone with access to the application being able to perform the same.

V. RESULT AND EVALUATION

We were successfully able to create an easy-to-use desktop application for the college facility to automate the process of result extraction and generation of excel sheet reports. The overall tedious time effort of hours together has been reduced down to 5-10 minutes of effort by building this ecosystem.

The application was tested with recent results released on university domain and provided accurate results with successful generation of the excel sheet report without overwhelming and time-consuming manual data entry methods.

The generation and processing time for results of about 60 students takes roughly about 4-5 mins, this is with stable internet speed and response time of the server, the time may vary with network speeds and request-response times of the results server.

VI. CONCLUSION

Based on results, this project concluded that University results batch processing system is more efficient and better way to handle cumbersome manual data entry problem. The results and analysis depict that users prefer to use a university results batch processing system rather than the manual system because it offers many features and functionalities which is otherwise available in the former system. It can be argued that our software system improves result analysis since it offers better ways of handling the numerous procedures in the manual preparation of excel reports. This is due to the users' greater confidence in the system. It also helps to eliminate the human error and accuracy when doing long tedious tasks like data entry.

VII. FUTURE SCOPE

The application has wide scope for improvement in the future wherein we can include various visualisation Graph modules for displaying charts and graphs for visual representation of extracted and processed data. More export formats

can be provided to the end user to generate and save the results. Charts and Graphical representation can also be included in template for excel using native excel charts API. The data that is collected across the years can be used as dataset for implementing machine learning models and predictions of results of department, subject or student based on the past performances and data inputs. Bots can also be configured to update the results manually when the results of the student are updated after reevaluation results are released by the university. More excel template options can be provided or options can be given for the user to customize the excel sheet template as per his requirement.

REFERENCES

- [1]. Gauripriya Kakati, Komal Revankar, Pratidnya Kittur, 2022 "WEB SCRAPING FOR VTU RESULTS", International Journal of Current Engineering and Scientific Research (Ijcesr)
- [2]. S. John W. Fowler, Lars Mönch, A survey of scheduling with parallel batch (p-batch) processing, European Journal of Operational Research, Volume 298, Issue 1, 2022, Pages 1-24, ISSN 0377-2217, <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