International Journal of Recent Advances in Multidisciplinary Topics
Volume 4, Issue 4, April 2023
https://www.ijramt.com | ISSN (Online): 2582-7839

Smart Scholarship Portal Using Natural Language Processing and Scraping

Prashansa Gupta[1*], Shashikant Solanki[2], Yash Gupta[3], Nidhi Mahto[4], Krishna Kant Singh[5]
[1,2,3,4]Student, Department of Computer Science and Engineering, Amity University, Noida, India
[5]Professor, Department of Computer Science and Engineering, Amity University, Noida, India

***Abstract*: Scholarship is an important aid and idea which basically helps in facilitating education for eligible students, especially those who came from socially and economically challenged backgrounds in the state, and bring them on to the mainstream for development. As we are in the world of technology and digitalisation, we need all of our services to be done quickly and digitally. In this paper, a chatbot enabled smart scholarship portal was designed and implemented using modern technologies and tools in which basically students will get all required information about all scholarships on a single portal. This smart scholarship portal focuses on ensuring a quick process of scholarships to needy students. It provides digital solutions that not only save time, but also have a good user-friendly interface and sends notifications using SMS service and emails on the progress of applications in a timely and transparent manner.**

***Keywords*: Web scraping, NoSQL database, Chatbot, User interface, Web-portal.**

1. Introduction

As a densely populated nation, India annually produces more than 6 crore graduates from a wide range of socio-economic and educational atmosphere. Almost an equal percentage of students enroll in universities and organizations to pursue many degrees which will help in their job life. Students eventually start looking for good scholarships based on their qualifications, abilities, income, etc. Receiving scholarships would enable these students to perform even better.

Scholarships in the market are growing along with technology and market growth, but the data is not centralized. People must look in many different locations for reputable scholarships. Understanding different national and international scholarships will be made easier by addressing this issue. Additionally, it will motivate pupils to improve their merit to be granted these scholarships for a better future.

2. Problem Statement

To design a web portal where several National and International scholarships that would be based on study-field, merit, income etc. are shown up. The most recent scholarships being provided will be updated using this real-time data management.

3. Methodology

A. *Problems of Existing System*

After graduation, students hunt for scholarships to pursue higher education in India or overseas. As technology and the market increase, so do scholarships and other government programmes, but the information is not localized in a single area. In order to find appropriate scholarships, students must search/refer to numerous websites.

In order to solve this issue, we are developing a central place where students may find scholarships based on their merit, skills, money, etc.

B. *Analysis and Features of Proposed System*

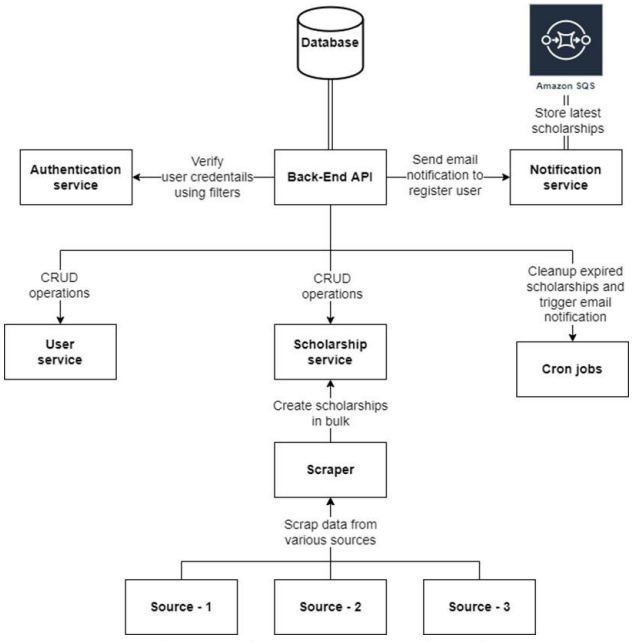

Fig. 1. Architecture of proposed portal

The portal is made using different combinations of various technologies and tools. Architecture of smart scholarship portal is divided into multiple components which are discussed further in paper. This portal will be having features of chatbots, notification services, good user-friendly interface and various other features which are required in this type of portal.

Various Components are:

*Corresponding author: prashansagupta43@gmail.com

a) Scraper
b) Notification Service
c) Database
d) Authentication Service
e) Scholarship Service

Components of the Web Portal:

1) Scraper:

A method for obtaining data from the World Wide Web (WWW) and putting it in a database or file system to be retrieved or analysed later is called web scraping. [1]. This is also called as web harvesting or web extraction. The data is routinely retrieved from websites with the help of HTTP or a web browser.

Portal will have an automated scraper which will scrape scholarships from various sources at scheduled time intervals and they will be stored in an apache kafka queue. Scholarships can be added to the database both ways by the scraper as well as the admin which will be in the JSON format.

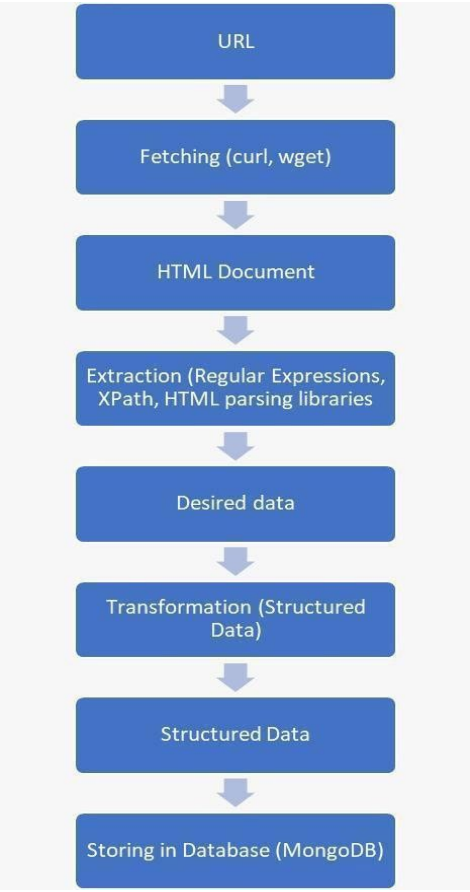

Fig. 2. Stages of proposed portal

As seen in figure, the web scraping project is primarily divided into three distinct stages, which are [2]:

a) Fetching stage: The fetching phase must be achieved before the target website with the required information may be accessed. The HTTP protocol is used for this, which is an Internet norm for sending and receiving requests from web servers. Web browsers access material on websites using similar methods. During this stage, one can employ libraries like curl 2 and wget 3 to transmit an HTTP GET inquiry to the specified destination address (URL) and acquire the HTML content as a reaction.

b) Stage of extraction: Important information should be extracted once the HTML page has been retrieved. Regular expressions, XPath queries, and HTML parsing libraries are employed in the extraction stage. A method for discovering information in documents is called XML Path Language (XPath).

c) Transformation stage: Having removed everything but the essential information, the data can now be organised into a layout that can be used for presentation or storage. The knowledge we may gather from the data that has been saved will help business intelligence make better judgments and much more.

Approaches to Web Scraping:

a) Regular Expression: A Python library called regular expression was initially created for the Perl programming language. Python uses the "re" module to handle regular expressions. When using a regular expression, we first define the pattern we wish to look for in a string, and then we look for any instances of that text that match. The pattern could seem odd because of the unique characteristics that alter how we interpret the pattern and the substance we intend to match.

b) Beautiful Soup: Leonard Richardson and a few other programmers created the Beautiful Soup Python package, which enables you to retrieve structured data from a website. XML and HTML parsing is performed using Beautiful Soup. Furthermore, it requires less steps to navigate, examine, and update a parse tree as compared to regular expression, making it considerably simpler to utilize. If the document doesn't mention an encoding, there is no need for you to keep a record of it because Beautiful Soup can transform incoming documents into Unicode and outgoing documents into UTF-8 automatically.

c) Lxml: Lxml is a library for XML and HTML processing that is widely regarded as one of the most comprehensive and functional tools available in Python. In addition, lxml supports XPath for extracting tree information due to its simplicity and ease of learning. To extract content fragments into a list, use XPath.

Additionally, if you've used CSS or XPaths previously, you won't have any difficulty learning this. Its inherent strength and quickness have also helped it become widely used in the sector. Although Lxml is an excellent tool for web scraping, some users choose not to use it since it can be challenging to install on particular PCs [7].

Table 1
Comparison of three approaches of web scraping

Scraping Approach	Installation Ease	Performance	Easy to use
RegEx	Easy (built in)	Good and Fast	No
Beautiful Soup	Easy	Little slow	Yes
Lxml	Moderate	Fast	Yes

It won't be a problem for you to employ a slower method like gorgeous soup if your scraping is limited to downloading data rather than obtaining it from a website. Regular expression is another fantastic option if you are only interested in scraping a small amount of data and don't require any additional dependencies or other parties. Overall, the lxml approach is the best since it is speedy and powerful, whereas the other two approaches are useful in certain circumstances only [9].

2) Cron Jobs:

A Linux tool called cron jobs is used to schedule tasks for later execution. It is often employed to schedule a routine task, such as sending a message every morning. You might need to set up cron jobs in order for some scripts, including Drupal and WHMCS, to carry out particular tasks.

Cron jobs will remove the expired scholarships from the database as well as trigger email notification to the user when a new scholarship is added to the database.

3) Scholarship Service:

Scholarship service includes:

- Creating scholarships in bulk
- Performing CRUD operations in database
- Auto-removal of expired scholarships.

4) Notification service:

Scraper will automatically scrape all the latest scholarships from the various sources and will add them to the database and admin also has the authority to add scholarships in the bulk. There are the two sources from which data can be added to the database. Using the notification service, the user will be notified every time a new scholarship is added to the database.

5) Authentication Service:

JSON web tokens are used to authenticate and verify user credentials.

JSON Web Token (JWT) (RFC 7519), an open standard, describes a simple and independent procedure for securely transferring data between parties as a JSON object. This information can be checked and trusted since it is digitally signed. JWTs can be signed using a public/private key combination using RSA or ECDSA, or with a secret key pair (with the HMAC algorithm).

6) Database:

Database will store all the scholarship data scraped from different sources. A NoSQL (MongoDB) database has been used which stores the data as key,value pairs. Scholarships have been stored in JSON format using the URL of the scholarships as the key [10].

4. Modules Description

The framework of the project is kept basic. The modules or libraries used in this project are highlighted below which includes description about them and how they have been integrated in our portal:

1) HTML:

Html (HyperText Markup Language) (HyperText Markup Language). It serves as the websites' base. Hypertext is the term used to describe hyperlinks that link web pages, either inside a single page or across many sites. An essential component of the Internet are links. It is responsible for designing the web page by displaying text, images, forms, tables etc. One can specify the layout and structure of the webpage by using the html tags.

2) CSS:

The Cascading Style Sheets or CSS is responsible for the web page appearance or styling. It is not a markup language rather it is a style sheet language. It is used to fastidiously style html or xml elements.

3) JAVASCRIPT (JS):

JS is the language that powers the web. It is the most well-known Scripting language. Brendan Eich created the first ever LiveScript (later named Javascript) at Netscape and today we cannot imagine the world without Javascript. It is supported by all major browsers like google, firefox, microsoft edge. JS has become an essential part of Front-end web development in the present era, giving developers with tried-and-true tools for creating scalable, interactive web applications.

4) Java:

Java is among the most extensively used and popular programming languages. It is an Object-Oriented Programming Language whose source code is first compiled into bytecode. Following that, the byte code is run on the Java Virtual Machine (JVM), apart from the embedded device.

5) Spring Boot:

The Spring Framework is the foundation for the Spring Boot project. It provides a faster and easier method to install, set up, and run both basic and web-based programmes. It is a Spring module that provides access to the RAD (Rapid Application Development) capabilities to the Spring Framework. It is used to build standalone Spring-based apps that may be executed since it requires very little Spring settings [14].

6) Java Mail API:

An API used to create, write, and read electronic communications is called JavaMail (emails). The JavaMail API offers a framework for sending and receiving emails that is agnostic of platform and protocol.

7) MongoDB:

MongoDB is a document database that internally stores BSON data but exposes itself as JSON. A document in MongoDB is a data structure made up of key value pairs, much like JSON objects are.

Web scraping is a logical match for a document database. Utilizing MongoDB eliminates the requirement to standardise data in order to fit the database. As an alternative, you can save the same objects that you use in code. Without creating a local database, MongoDB Atlas, the database-as-a-service solution from MongoDB, makes it simple to store data that has been scraped from websites [5].

The steps in storing data in database (MongoDB):

- *Build a spider for scraping a website:* There are classes in Scrapy known as spiders to state which pages to crawl and how to parse data on that particular web-page.
- *Build an Item:* Scrapy has features of items consisting of a dictionary-like API and some extra features to ease down the process of structuring data that would generally be unstructured.

- *Build an Item Pipeline:* Scrapy have Item Pipelines that are used in processing the items which have been scraped from particular web-page. Once an item is provided, it moves to all pipelines you've stated in the crawler configuration.

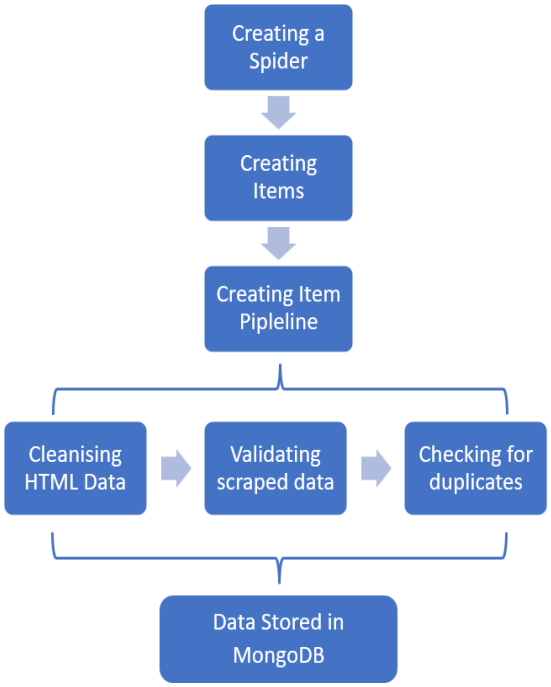

Fig. 3. Storing data in MongoDB

8) Amazon SQS:

With Amazon SQS (Simple Queue Service), you may send, store, and receive messages across software components in any number without worrying about message loss [14].

9) Heroku:

Heroku is a cloud - based platform that integrates computation, information, and workflows with a greater developer experience. Heroku Enterprise provides the same fantastic developer experience in addition to enterprise-level governance, collaboration, and compliance.

The platform offers corporate teams a quickest route to providing dependable consumer experiences at scale. It improved confidentiality in an internet runtime environment, seamless scalability to suit enterprise demand, and security. simpler complies with PCI, SOC, and other standards Automated, CI/CD workflows for the best team cooperation. [13]

10) Nuxt.js:

Nuxt is the Hybrid Vue Front-end Framework. For the creation of cutting-edge and effective web apps that can be installed on any platform supporting JavaScript, Nuxt is an open-source framework available under the MIT license. Vue.js is the view engine used by Nuxt. Component auto-imports and file-based routing are two capabilities that Nuxt adds to the frontend framework Vue. Users of Nuxt can now access new patterns thanks to the integration of Vue 3, the newest major release of Vue.

11) Vuetify:

A Vue UI library called Vuetify features exquisitely made Material Components. Its objective is to give developers the resources they need to create rich and interesting user interfaces. Vue is a name for a modern user interface framework.

Unlike other monolithic frameworks, Vue is built from the ground up to be adopted incrementally. Because it solely focuses on the display layer, the core library is simple to use and integrate with other libraries and ongoing projects.

In addition, when combined with contemporary tools and auxiliary libraries, Vue also has the ability of driving established and modern single page applications. We use vue.js in this portal since it is lighter and more customizable than other frameworks such as React.

12) Vercel:

Vercel is a frontend stack for web developers that isolates the backend and simplifies the development of websites and JavaScript-based applications.

It runs off the necessity to oversee a web server. It interfaces with your content or database and offers zero-configuration compatibility for 35+ frontend frameworks. Here, vercel is used with nuxt.js and vue.js framework.

Vercel is a cloud platform for serverless and static frontends. It makes it possible for programmers to host websites and web apps that scale automatically, deploy rapidly, and don't need any manual intervention.

13) ANN:

Artificial neural networks are a Deep Learning model that is specifically designed to interpret sequential input. This stage's input is the output from the step before it.

It uses the same parameters for each input to perform the very same operation on all the inputs or hidden layers to generate the output [17].

Most important applications of artificial neural networks are machine translation and natural language processing [6].

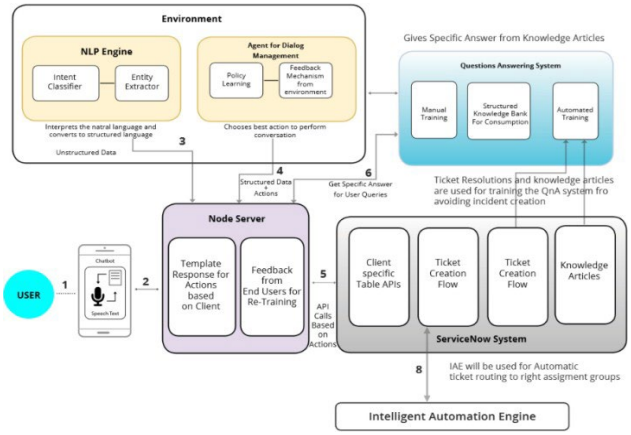

Fig. 4. Chatbot architecture

Components of conversational chatbot architecture:

1) Environment: This is the location where one can locate context interpretation and the NLP engine.

NLP Engine: The central element of the system that translates what users say at any given moment and transforms

the language into structured inputs for the system to process further is the NLP Engine. The chatbot must handle a large number of features because it is domain- specific. Advanced machine learning algorithms and Neural Networks Algorithms are used in the NLP engine to recognise the user's purpose and then match it to the range of possible intents the bot supports. [4]

NLP Engine further has two components:

a) Intent Classifier: This function examines the user's input to identify its significance and associate it with one of the chatbot's supported intentions.

b) Entity Extractor: This programme is responsible for obtaining important data from the user's query.

2) Question and Answer System: This is essential for giving customers answers to questions that are commonly asked or inquired about. This system analyses the query and then returns suitable knowledge base answers. It consists of the following elements:

- Manual Training
- Automated Training

3) Plugins/Components: Plugins are used to provide chatbots with APIs and other advanced automation components.

4) Node Server/Traffic Server: The server that manages user traffic requests and directs them to relevant parts. The answer from internal components is also forwarded by the traffic server to the front-end systems.

5) Front-End Systems: Front-end systems refer to any platforms that interact directly with clients or end-users. In this project, it is the scholarship portal which is the front-end system.

5. Conclusion

In this work, a chatbot enabled smart scholarship portal was designed and implemented that basically replaces the manual methods. This online web-based portal helps students to apply for scholarships through the internet irrespective of their location around the globe. It is basically implemented in such a way that many loopholes or issues which exist in other portals and applications have been eliminated to a larger extent.

6. Future Work

For future work, more features can be added to enhance the user experience on this portal. Features like face recognition for login service, recommendation system and automation can also be integrated to this which will definitely benefit the students.

References

[1] Jiahao Wu, “Web Scraping using Python: Step by step guide” ResearchGate publications (2019).
[2] Matthew Russell, “Using python for web scraping,” No Starch Press, 2012.
[3] Ryan Mitchell, “Web scraping with Python,” O’Reilly Media, 2015.
[4] Shah, Seema. (2019). A Comparison of Various Chatbot Frameworks. Journal of Multi-Criteria Decision Analysis.
[5] Ali, Wajid & Majeed, Muhammad & Raza, Ali & Shafique, Muhammad Usman. (2019). Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. Asian Journal of Computer Science and Information Technology. 4. 1-10.
[6] Tamrakar, Rohit & Wani, Niraj. (2021). Design and Development of CHATBOT: A Review.
[7] Thomas, David & Mathur, Sandeep. (2019). Data Analysis by Web Scraping using Python. 450-454.
[8] Zhao, Bo. (2017). Web Scraping.
[9] Kasereka, Henrys. (2020). Importance of web scraping in e-commerce and e-marketing. SSRN Electronic Journal.
[10] Krishnan, Hema & Elayidom, M. Sudheep & Santhanakrishnan, T. (2016). MongoDB – a comparison with NoSQL databases. International Journal of Scientific and Engineering Research. 7. 1035-1037. Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80.
[11] Salehinejad, Hojjat & Sankar, Sharan & Barfett, Joseph & Colak, Errol & Valaee, Shahrokh. (2017). Recent Advances in Recurrent Neural Networks.
[12] Danielsson, Patrik & Postema, Tom & Munir, Hussan. (2021). Heroku-Based Innovative Platform for Web-Based Deployment in Product Development at Axis. IEEE Access. pp. 1-1.
[13] Hernández, Sergio & Fabra, Javier & Álvarez, Pedro & Ezpeleta, Joaquin. (2013). A Reliable and Scalable Service Bus Based on Amazon SQS. 8135. 196-211.
[14] K. Guntupally, R. Devarakonda and K. Kehoe, "Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example," 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 5328-5329.
[15] Devi, Jyoti & Bhatia, Kirti & Sharma, Rohini. (2017). A Study on Functioning of Selenium Automation Testing Structure. International Journal of Advanced Research in Computer Science and Software Engineering. 7. 855-862.
[16] Alshammari, Reem & Alrashed, Rwan & Almutiri, Atheer & Alwalah, Mathail & Al-marrai, Wadha & Alqahtani, Dana & Alzahrani, Amani. (2021). Data Extraction Based on Web Scrapy.
[17] Dhyani, Manyu & Kumar, Rajiv. (2020). An intelligent Chatbot using deep learning with Bidirectional RNN and attention model. Materials Today: Proceedings.