A digital approach to access and manage distributed data

Prof. Sargar D. Korde
Information Technology
K.J. Somaiya College of Engineering
Mumbai, India
sagar.korde@somaiya.edu

Nishith Khandor
Information Technology
K.J. Somaiya College of Engineering
Mumbai, India
nishith.k@somaiya.edu

Ayush Mittal
Information Technology
K.J. Somaiya College of Engineering
Mumbai, India
ayush.mittal@somaiya.edu

Zirak Mistry
Information Technology
K.J. Somaiya College of Engineering
Mumbai, India
zirak.mistry@somaiya.edu

Dhruv Mehta
Information Technology
K.J. Somaiya College of Engineering
Mumbai, India
mehta.dh@somaiya.edu

Abstract—With the advent of Information Technology in todays time, there is no dearth of increasing requirements for storing and retrieving data. As the data produced per year increases at an explosive rate, storing this data and representing it in a well structured format and in a timely manner is one of the biggest issues today for the organizations and institutions. This paper proposes an approach to manage and access large amounts of distributed unstructured data of an organization in an efficient manner. Hence, a web-application has been designed to efficiently access and manage the data and to save precious time spent in accessing the distributed data. Also the paper proposes the methodology which describes the features provided by the web-application.

Index Terms—unstructured data, nosql, python, distributed, excel, upload, mongodb

I. Introduction

Today, in most of the institutions and organizations the major problems relating to the maintenance of physically documented data and their storage is becoming more and more cumbersome and labour intensive. Many institutions use physical documents to manually feed in the data. The amount of time and resources spent in creating, storing and maintaining this physical data is an issue of concern. This major problem relating to physical data can be divided into the following categories:

Data availability- The very nature in which this data is created and stored, does not provide the staff of the organization with the required data, based upon when and where it is needed.

Data accessibility- The access to the data by the staff is also a tedious task. This task may involve a series of permissions to be taken from other staff members of the institution. So, even if the data is available, it is always not necessarily accessible.

The crux of the matter is that the data is not available and accessible at all times to the authorized individuals.

Providing a solution to these problems is very important for the organizations and institutions so that they can reduce the time spent on data retrieval, reduce physical effort of the employees of the organization or institutions, reduce or eradicate the human errors involved in manual entries and spend more time on the productive work.

A few decades ago, these problems were considerably difficult to circumvent due to the limited expertise and technological limitations. In todays time, considering the technological advancements in the technological sector, most of these problems can be solved to a great extent. An ERP system can prove to be a solution to the above problems. At the core, this system will integrate, organize, and standardize the data of the organization or institution. The system will be hosted on the internet. As a result the ERP system will provide the following:

1) Data availability by providing the data to the authorized individuals on the internet. Since the data is hosted on the internet, the data will always be available to the users.
2) Data accessibility to the authorized users by a simple authentication mechanism. The users can access the data whenever they wish to simply by authenticating themselves.

A. Structured Data vs. Unstructured Data

Why did we use unstructured data?What are the problems of we use structured data?
Structured data is a type of data that has a fixed format and follows a predefined structure of storage and predefined data model. The structured type of data is usually stored in relational databases(RDBMS) and data warehouses. Data such as phone numbers, Social Security Numbers, or ZIP codes are stored into the form of length-delinated data.

Also string in the form of variables like names are stored in the relational databases. This type of data can be human or machine generated. This well structured format of data storage makes it very easy to access the data with the help of structured queries also called as SQL queries.

Unstructured data is a type of data that may not have a fixed format and no pre-defined data model. These characteristics of unstructured data allows the different types of data to be stored. These types of data can be text, videos, sound or other formats. On the other hand these characteristics make data retrieval and search difficult. Unstructured data is typically stored in NoSQL databases, data lakes and data warehouses. Again this type of data can be human or machine generated. The type of queries that are required to access this data are called unstructured queries or NoSQL type of queries.

Why this project utilizes unstructured form of storage?
Unstructured form of storage is considered for this project for many reasons. The system under consideration requires considers the following aspects-

speed,
efficiency
scaling
adaptibility

The first reason for considering this type of storage is that the considered ERP system requires that the data be retrieved at a quick pace and in an efficient manner. The second reason is that the data format for the system may change over time. So the data may have to adapt to the system requirements in the future. The third reason is that the data may increase in an linear or exponential manner in the near future.

All of these requirements are satisfied by unstructured form of storage. The first requirement of quick retrieval of data is satisfied as the querying is pretty fast and efficient. The second requirement is adaptability. Since the data model is not predefined hence the change in the storage format or the data model can be incorporated in the system quite easily. The third requirement that is satisfied is scalability. The system that have the ability to handle unstructured data can scale-up. Scaling-up, which is also called vertical scaling involves in deploying faster serves with higher processing speeds and much more memory.

Why would structured prove to be incompatible with the systems requirements?

Due to the very nature of structured data storage the system requirements will surpass what the data model will be able to provide. Following are a series of reasons why structured form of storage will prove to be incompatible with the system requirements:

1) The data model of the system may need to be changed multiply times during the life-cycle of the project. Tak-ing this aspect into consideration, structured type of storage cannot perform alterations in its data model as it is well-defined. Hence, structured data does not support adaptability.
2) The data of the system can increase at a fast pace or also increase exponentially. Unstructured type of storage cannot keep up with the increased demands for data storage as it lacks the characteristics of big data. Hence, structured data does not support scalability.
3) The queries of NoSQL databases is much more easier to frame and execute as opposed to the structured queries or SQL queries.

B. Python vs. Java

Why python is better than java?
Java is a compiled language whereas python is interpreted language. Python can perform the same function as of Java in fewer lines than java. There is not much configuration required for python flask but any java framework(example; Spring MVC) requires lot of configuration and boilerplate code.

C. MongoDB support in java and python

The mongoDB driver for Java seems to be directly derived from JavaScript but the usability and efficiency suffers a bit because Java does not have literals for maps/objects like JavaScript does. Whereas python does have literals for maps. Hence python has an advantage over java for MongoDB.

Python supports efficient metaprogramming through powerful introspection features that allow programs to inspect and modify code at runtime. Java's introspection is quite poor and inflexible and thus metaprogramming requires writing preprocessors resulting in increased cost and complexity of implementing the ERP5 system.

Python can be used for both scripting and core development thus reducing complexity and increasing the flexibility of the system whereas java can only be used only for core development. With java,it becomes necessary to provide a separate scripting environment based on different language such as Jython or ECMAScript in order to allow flexible configuration at run time by ERP administrators

D. Web Technologies

The web technologies that are used for the development of the system are HTML, CSS, Bootstrap, JavaScript along with the python MVC framework Flask. Following are the reasons for the use of the above mentioned technologies:

1) The basic web technology HTML, will provide the web based ERP system with the required MarkUp for the data to be displayed in the form of web pages.
2) The basic styling technology called CSS(Cascading Style Sheets), will provide the webpages with the re-quired styling and an overall customized look of the system in the form of user-interface.

3) The rich library of Bootstrap with predefined styling and themes will add on to the basic CSS. Also the biggest advantage of using Bootstrap for the ERP system under consideration is to make the UI very responsive over many platforms.
4) The power scripting language JavaScript will help to process request that come from the client side. As the system user will be requesting for data most of the time so, handling those requests is most important.
5) An efficient and power MVC framework based on python is the Flask framework. This framework proves to be ideal for the systems development because it works well with the NoSQL databases which the system uses and as it is based on python, writing code is much more easier.

II. Related Work

In [1] a architecture is proposed to integrate multiple databases and provide an access which is transparent to repositories.Cross database retrieval uses the metadata to integrate the data and achieve interoperability which is good for querying. The architecture of the system is layered and the design follows principle of modularity and low coupling to achieve good scaling and compatibility. But it does not provide away of accessing the unified source with a good user interface and does not implement any security standards required for the management of the unified data.The architecture gives the output in standard XML format and the user has read only access and hence cannot interact and update the data.

In [2] the author explains how the data acquisition or retrieval takes place when real-time monitoring systems are concerned. The authors contrast many different C++ frame-works which use the technology of multithreading and the concept of ring-buffer data structure on basis of the time interval between the moment when the data is produced and the time when the data is collected. Although C++ is a good language as far as simplicity is concerned, it is not very secure, prone to many errors, and is platform dependent. Instead of C++, the data acquisition systems can be implemented in much better languages such as java or python which are more secure and robust.

III. Methodology

The project is designed as a web based system to provide high accessibility, remote access and availability of data.

The web based system uses appropriate access control system for accessing and updating the data on the system. The students can only view their data. The faculty and staff can view data of any student and upload data in accordance with the provided access rights. The faculty teaching a subject can upload the term test marks and practical marks of that subject. The data of newly admitted students which will contain students name, caste, category and other basic details which will be required by institution will be uploaded by admissions cell. The placement cell will upload the students internship details, certificates and placement details. The examination department will upload the End Semester exam marks of student. The different college committees faculty advisor will upload the list of students that are part of the committee. The account of the faculty is approved by the hod/respective administrator before the user can login.The system allows the records of students to be frozen according to the year of admission making the frozen records uneditable by any user. The system is built for educational institutes to move from the traditional approach to a ERP approach with unified system to store and access the data as and when required. The database will store the entire details of the student from admission to the academic data such as students marks and extra curricular and cocurricular achievements such as internships, member of any organisation and committee, any competitions and certificates. The web site consists of user friendly layout to read the data and the update of the data is done with the help of excel sheets which are converted to csv formats before storing in the database. The framework used is the flask microframework with support of python and pymongo as the language to access the MongoDb database used in the application. Python supports the necessary libraries required for the application. The application features the support for storing the data on the cloud with the google drive cloud service. The data supported for cloud storage are pdfs and other necessary documents.

The application provides the user with search of a record based on the name, roll number and email address of the student. The search requests are fulfilled using ajax call made to the server and the response is displayed in the format of a table which is easy for the user to read and distinguish the necessary details. The filters are provided with easy search of the necessary details for the user. The filters are based on department of the student, the current year of the student and the cgpa of the student and others. The application supports printing the data displayed on the web page as a document in user readable format. The data can be visualized as graphs in form of bar graphs. The data of admissions based on the numbers of admissions in the past five years, the numbers of students in different categories and branches for the current Year can be visualized. The data uploaded in the excel format is dynamic in the way the application is able to map different column names to the required keys in the database. This is achieved by checking the different supports the column headers with different names than specified which is searched by the system in the background with the use of permutations and combinations of the desired name in the header. The header is then mapped with the actual header and if no match is found then the uploaded sheets needs to be changed in the headers by the user.

An additional feature of the system is that, when the data is uploaded in excel format the pre-defined header strings

of the system will be compared to the permutations and combinations entered in the corresponding headers of the excel sheets by the faculty. If a particular header of the excel sheet does not match will the corresponding pre-defined header string of the system for that column of data, then the header of the excel sheet is considered invalid and the faculty will be requested to re-upload the excel sheet. Hence, the faculty will have to provide the valid header name for that column of data and then re-upload it.

The new feature of the system is the student of the year. This feature of the application will provide the user will a list of students who have performed excellent throughout the year. This feature works in the following manner:

1) The system will be defining the basic criteria for the students. The criteria will be providing the weightage for different categories, with academics having a weightage of 40%, internships with a weightage of 30%, and extra-curricular activities with a weightage of 30%.
2) The score for each student will be calculated based on the mentioned categories and their respective weigh-tages.
3) All the scores will be sorted and the top 10 students will be recommended by the system on user request.

The application will also provide a feature to freeze the data of the alumni students. This data of the passed out students also called as alumni students will be frozen. In other words, the data of these types of students will be finalized and no changes to the data will be allowed by any alumni.

IV. Results


```
"_id" : ObjectId("5c288093ad70863dc4e59ae1"),
"roll_no" : 1,
"name" : "a",
"email" : "a@abc.com",
"branch" : "COMS",
"current_year" : "F",
"Caste" : "Open",
"HandiCapped" : "No",
"Gender" : "M",
"Minority" : "Sikh",
"Other_category" : "No",
"cgpa" : 9.96,
"placement" : [
        {
                "Non-Dream" : "-",
                "Package" : "-"
        },
        {
                "Dream" : "-",
                "Package" : "-"
        },
        {
                "Dream" : "-",
                "Package" : "-"
        }
],
```


Fig. 1. ”student details unstructured”

Please refer Figure 2 for faculty home page which renders the unstructured data shown in Figure 1. Unstructured storage helps in storing multiple nested values with ease. It is an extended version of the student home page where the faculty will be able to view the data of all the students in the database as opposed to the student home page where the student will only be able to view his data and no other students data. Also, the faculty will be provided with the search option along

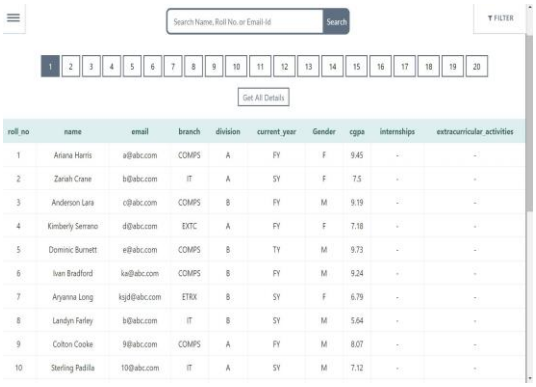

Fig. 2. ”faculty home page”

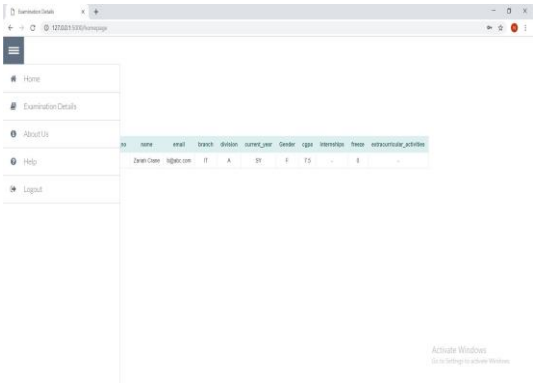

Fig. 3. ”student home page”

with the filtering functionality. The filtering functionality with its pre-defined filter strings will allow the faculty to filter out the student data depending upon the requirements. Please see Figure 3 for student home page layout. The students after logging in to the application will be directed to their homepage. On the homepage, they will be able to see all the data pertaining to them such as personal details, academic details, extracurricular and co-curricular details in sectional format.

Please refer Figure 5 for student examination details. The Figure 4 shows the unstructured format for storage of the examination details in the database. Referring to the unstructured data the data is highly nested which is rendered to a structured format for ease of view to the user. The student can click on examination details option in navigation bar to view his/her detailed examination records. The detailed examination records will show the student his/her marks per subject, final gpa per year and overall cgpa of all his/her semesters.

The graph data will provide the information to the user in the form of data visualization technique which will be in the form of bar graphs. These bar graphs will provide a representation of the admission data over the passed five years. The total admissions for a particular year irrespective of the departments will be represented by a single bar graph.Figure 6 is an example of graphical view of the students in different

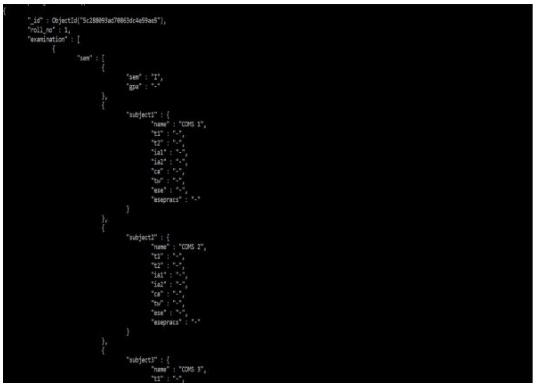

Fig. 4. "student examination details unstructured"

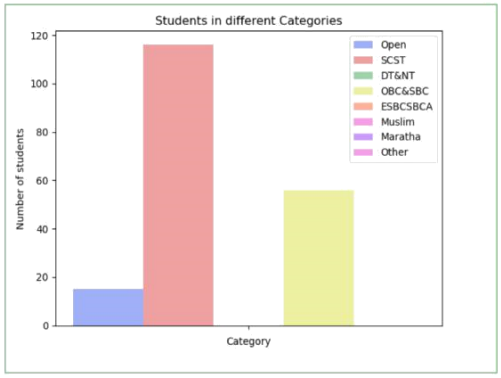

Fig. 6. "category graph plot"

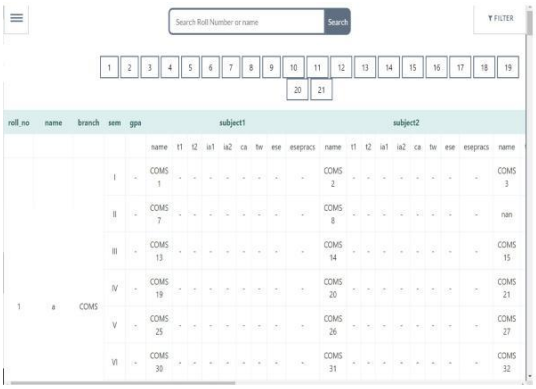

Fig. 5. "student examination details"

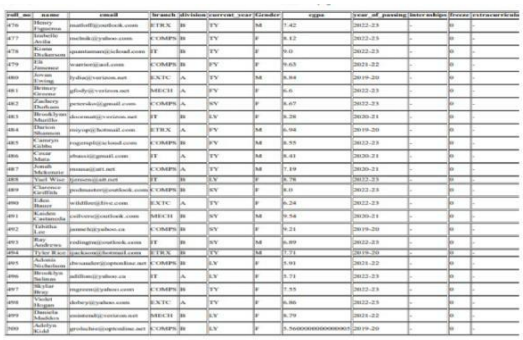

Fig. 7. "save as pdf"

categories during a particular admission year.

The print table will provide the feature of exporting the current table data into a PDF document.The Figure 7 shows a table being saved as pdf format. The page allows to select from saving the document and even sending the request to print the document.

V. Conclusion

The design and the different functionalities of the ERP system for this application has proved to be an ideal approach for satisfying requirements for the organizations and educational institutions. The results and outputs generated by the system have successfully solved the data problems such as data availability and data accessibility. The system has also provided the over requirements such as speed, efficiency, scalability and adaptability. Also this system will provide ample room for further advancements of this system in the future. So, in conclusion a system like this can can keep up with the organizational needs and will go a long way in helping the educational institutions in integrating, retrieving and maintaining their students data.

VI. Future Work

Currently the application provides a scalable solution for merging and accessing the distributed data of Educational Institutions remotely. Certain facilities can also be added to make it more useful and efficient.:

1) The application can provide a end to end encrypted channel for transmission of data between the server and the client in a secured way.
2) We shall incorporate data analytics into the application to get some valuable statistics about the students.
3) The application can be generalised to use it for Enter-prises. We shall add more functionalities to the system to convert it into full fledged Enterprise Resource Planning or Customer Relationship Management Systems.

References

[1] Jiaman Liu, Xu Du, Hao Li, Juan Yang, "Heterogeneous Learning Resources Integration and Cross-Database Retrieval", International Conference of Educational Innovation through Technology (EITT), 2017.
[2] Rolando Ingls, Piotr Perek, Mariusz Orlikowski, "A simple multithreaded C++ framework for high-performance data acquisition sys-tems", Department of Microelectronics and Computer Science, Lodz University of Technology, Lodz, Poland, 2015.
[3] https://www.infoq.com/articles/mongodb-java-php-python
[4] https://getbootstrap.com/docs/4.1/getting-started/introduction/