STUDENT DATA INTEGRATION AND CONSOLIDATION SYSTEM

By

NISHITH KHANDOR * DHRUV MEHTA ** ZIRAK MISTRY ***

AYUSH MITTAL **** SAGAR KORDE *****

*-***** K.J. Somaiya College of Engineering, Mumbai, Maharashtra, India.

Date Received: 23/04/2019 Date Revised: 02/06/2019 Date Accepted: 30/06/2019

ABSTRACT

With the advent of Information Technology in today's 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. This system also makes an attempt to solve the problem of data availability and data accessibility, both of which are very important from an organization's point of view that works with data on a day-to-day basis. Data security for organizations is very important and hence the system also incorporates a security feature with the help of the SSL technology, which will ensure that data transfer remains private and integral over the internet during the client and server communication. Also the paper proposes the methodology which describes the basic features provided by the web-application along with a set of special features that can help reduce effort and provide useful analysis and data visualization based on the data of the organization.

Keywords: Unstructured Data, NoSQL, Python, Distributed, Excel, Upload, MongodB.

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 labor 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 today's time, considering the technological advancements in the technological sector, most of these problems can be solved to a great extent. An

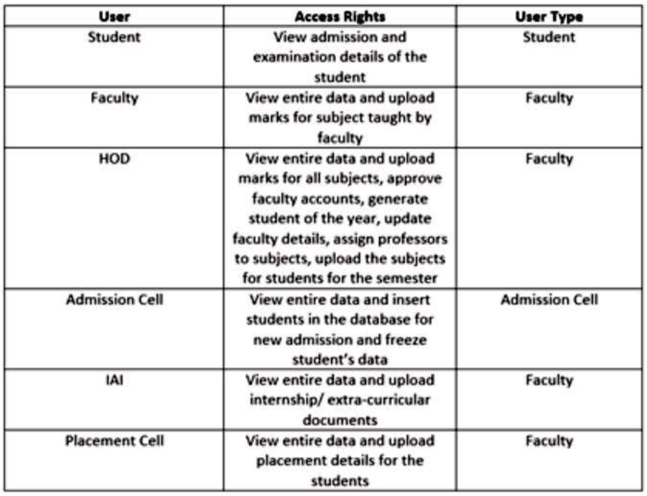
ERP system can prove to be a solution to the above problems. At the core, this system will integrate, organize, and standardize the data of the organization or institution. The system will be hosted on the internet. As a result, the ERP system will provide the following,

- 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.
- 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.

The users of the system will have different roles. Following are the different users the system consists of,

- Student
- Faculty
- Admission Cell

Since there are many users, each with their user rights hence, they will also have access rights to the system. This is important from a security standpoint. No user will be given full access to the system's data. Figure 1 shows the access rights to the systems data as per the user.



User	Access Rights	User Type
Student	View admission and examination details of the student	Student
Faculty	View entire data and upload marks for subject taught by faculty	Faculty
HOD	View entire data and upload marks for all subjects, approve faculty accounts, generate student of the year, update faculty details, assign professors to subjects, upload the subjects for students for the semester	Faculty
Admission Cell	View entire data and insert students in the database for new admission and freeze student's data	Admission Cell
IAI	View entire data and upload internship/ extra-curricular documents	Faculty
Placement Cell	View entire data and upload placement details for the students	Faculty

Figure 1. Access control table

Structured Data vs. Unstructured Data

Why did we use unstructured data? What are the problems if 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 (SSN), or ZIP codes are stored in the form of length-delineated data. Also string in the form of variables like names are stored in relational databases. This type of data can be human or machine generated. This, well-structured format of 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 searchdifficult. 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 the following aspects.

- Speed
- Efficiency
- Scaling
- Adaptability

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 a 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 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 structure 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:

- The data model of the system may need to be changed multiple times during the life-cycle of the project. Taking 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.
- 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.
- The queries of NoSQL databases is much easier to frame and execute as opposed to the structured queries or SQL queries.

Python vs. Java

Why python is better than Java?

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

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 meta programming through powerful introspection features that allow programs to inspect and modify code at runtime. Java's introspection is quite poor and inflexible and thus meta programming 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 Python or ECMA Script in order to allow flexible configuration at run time by ERP administrators.

Web Technologies

The web technologies that are used for the development of the system are HTML, CSS, Bootstrap, and JavaScript along with the python MVC framework Flask.

Following are the reasons for the use of the above mentioned technologies.

- 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.
- The basic styling technology called CSS (Cascading Style Sheets), will provide the webpages with the required styling and an overall customized look of the system in the form of user-interface.
- 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.
- 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, handling those requests is most important.
- 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 easier.

1. Objective of Study

The objective of the study is to find a solution to bring the data distributed across organization at one place and provide an ease of access to this data. The data should be easy to visualize and the proposed system should contain access rights similar to those of the current system.

2. Related Work

Liu, Du, Li, and Yang (2017) have proposed heterogeneous Learning Resources Integration and Cross-Database Retrieval. An architecture was 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 a way 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.

XML format has some disadvantages, i.e. syntax of XML is verbose and redundant compared to JSON like other text-based data transmission formats.

When the volume of the data becomes large, this redundancy causes higher storage and transportation costs. Compared to JSON like other text-based data transmission formats, XML is less readable and arrays are not supported in XML.

Inglés, Perek, Orlikowski, and Napieralski, (2015) have presented a simple multithreaded C++ framework for high-performance data acquisition system. They explain how the data acquisition or retrieval takes place when real-time monitoring systems are concerned. They contrast many different C++ frameworks, which use the technology of multi-threading 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.

Lee and Zheng (2015) have studied about revitalizing information management systems in Afghan settings through not only SQL (MongoDB) technology.

Gharanai, Gh, and Ahmadi (2016) explained that structured Databases cannot handle huge data efficiently and effectively. Therefore, they propose the use of unstructured database management system namely MongoDB to replace the structured databases and to solve the issue of efficiency. They also gave an overview of MongoDB and explains its various features such, as scaling out, sharding, and replication. Further, the author compares MongoDB and MySQL on the basis of its features and CRUD operations. MongoDB is faster than MySQL since it took only 0.5 seconds to insert 20000 users, whereas MySQL took 880 seconds. They conclude by encouraging developers to use and promote non-relational databases paradigm due to its features such as scalability and availability, which are a must for Big Data Organizations.

Hanine, Bendarag, and Boutkhoum (2016) analyzed the Post-Implementation Evaluation of a University Management Information System (UMIS).

Symeonaki, Papoutsidakis, Tseles, and Sigala (2016) conduct a post-implementation evaluation of a Student Information System at Piraeus University of Applied Sciences based on D&M IS success and Technology Acceptance (TAM) models and Wixom and Todd Framework. A thorough evaluation of the system was conducted by processing and analyzing the research objectives scores, the comment responses, and the respondents' demographic profiles. The concerned Student Information System consists of modules, such as Students Records - Secretariats Management, Student Club, Personnel Management, Financial, and Assets

Management, Procurement Management, Electronic Protocol and Workflow Management, and Information subsystem. The data for evaluation of system was collected using questionnaire provided to a small group of users. The parameters used for IS Success model is Information Quality and System Quality and those used for TAM model are Usefulness, Ease of Use Intended use and Experience. They have used Cronbach Alpha technique for reliability analysis and face validation for validation analysis. The outcomes of this research suggest that PUAS and its academic community could be greatly benefitted by the implementation of the Student Information System in terms of strategy planning, distributing resources as well as organizational success. The system improves the provided services to students and minimize their contact with the Secretariats of their Departments.

Babu and Surendran (2017) explain the relational to NoSQL Database Migration. They speak about how relational databases are not able to keep up with the exponential increase in the data storage requirements that can be of the order of terabytes and petabytes. Today most of the data is in semi-structured or unstructured format. With a large variety of data that needs to be stored, there are a lot of challenges to be faced with storing such data, which does not find solution in RDBM systems. The author encourages the use of NoSQL databases and contrasts three NoSQL type databases namely,

- MongoDB
- Cassandra
- Hbase

The comparisons between these NoSQL databases are based on the following characteristics possessed by each of the databases. Following are the aspects under which the comparisons have been performed.

- Key Characteristics
- Operating Mode
- CAP theorem (Consistency, Availability, and Performance)
- Applications
- Indexing
- Distribution
- Usage case

The author provides the following migration methodologies from RDBMS to NoSQL.

- Migration based on etl (extract, transform and load) (Hanine, Bendarag, & Boutkhoum, 2016)
- linked list based migration (Lee & Zheng, 2015)
- graph based migration (Zhao, Lin, Li, & Li, 2014)
- a framework for migration and query conversion (Rocha, Vale, Cirilo, Barbosa, & Mourão, 2015)

Veen, Waaij, and Meijer (2012) compared Cassandra and MongoDB and proved that MongoDB does perform better than Cassandra when using a single server instance. On the other hand, Cassandra is considered because of its horizontal scalability.

3. Methodology

The research work 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 student's 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 student's internship details, certificates, and placement details. The examination department will upload the End Semester exam marks of the student. The different college committee's 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 non-editable by any user. The system is built for educational institutes to move from the traditional approach to an 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 student's marks and extra-curricular and co-curricular achievements such as internships, member of any organization and committee, any competitions and certificates. The website 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 micro framework with support of Python and Pymongo as the language to access the Mongo DB 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.

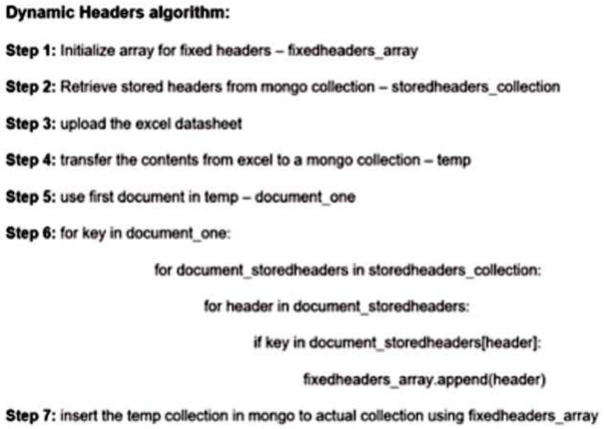
The data can be visualized as graphs in form of bar graphs. The data of admissions based on the numbers of admissions in the past five years, the numbers of students in different categories and branches for the current Year can be visualized. The data uploaded in the excel format is dynamic in the way 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 need 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 with 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 with 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. As mentioned above, the new feature of the system is the dynamic headers concept. The dynamic headers will allow the faculty to enter the data without worrying about the format of each column. Flexibility is given to the faculty to enter the data into the excel sheets with respect to the name of the headers as well as those headers in different columns. Refer Figure 2 for the concept of dynamic headers.

The new feature of the system is the student of the year. This feature of the application provides the user with a list of students, who have performed excellently throughout the year. This feature works in the following manner.

- All the scores will be sorted and the top 10 students, will be recommended by the system on user request.
- 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 extracurricular activities with a weightage of 30%.
- The score for each student will be calculated based



```
Dynamic Headers algorithm:
Step 1: Initialize array for fixed headers – fixedheaders_array
Step 2: Retrieve stored headers from mongo collection – storedheaders_collection
Step 3: upload the excel datasheet
Step 4: transfer the contents from excel to a mongo collection – temp
Step 5: use first document in temp – document_one
Step 6: for key in document_one:
            for document_storedheaders in storedheaders_collection:
                for header in document_storedheaders:
                    if key in document_storedheaders[header]:
                        fixedheaders_array.append(header)
Step 7: insert the temp collection in mongo to actual collection using fixedheaders_array
```

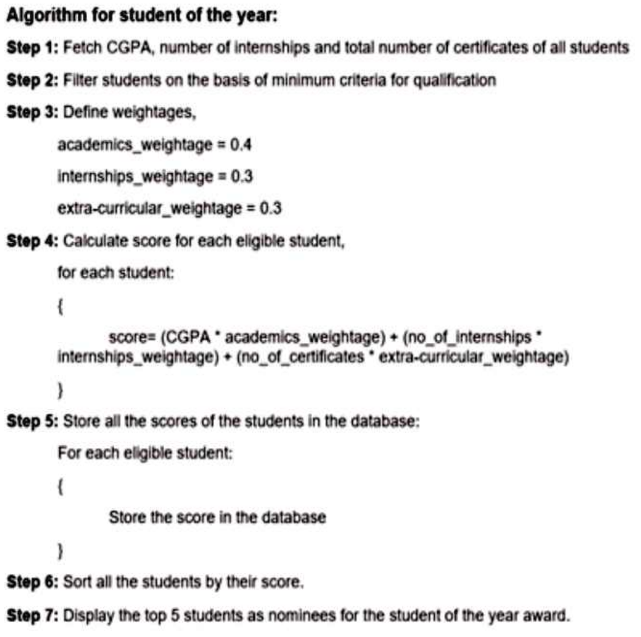
Figure 2. Dynamic Headers Algorithm

on the mentioned categories and their respective weightages.

Figure 3 shows the proposed algorithm for "Student of the year" functionality. 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.

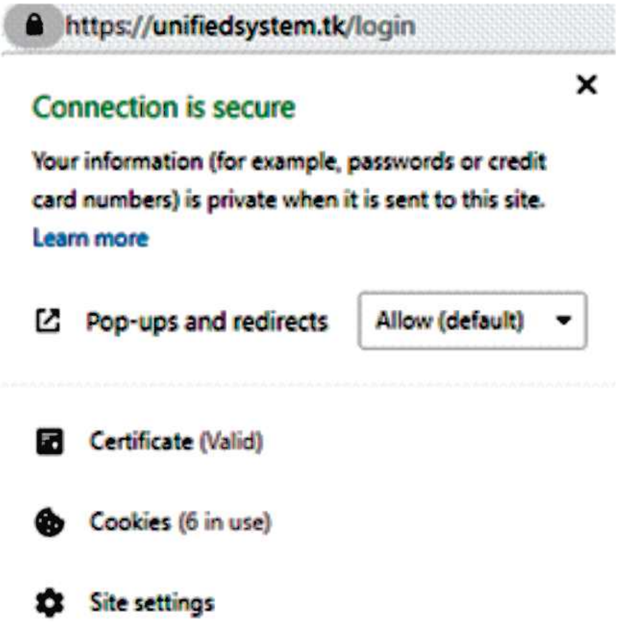
From the perspective of security, the system is also implemented with SSL for the web-application to provide trust certificate and end-to-end encryption between the client and server and to prevent man-in-the-middle attacks. It also helps to increase the SEO score of web-application. We have used Let's Encrypt, a free, automated, and open Certificate Authority (CA) to generate the certificate and private key. Certbot is a Automatic Certificate Management Environment (ACME) client for generating SSL certificate using Let's Encrypt. Let's Encrypt will bind the generated certificate and private key to a domain name. For this web-application, the authors have used AWS EC2 to host the application and freenom.com to get free domain name.

The screenshot of the secure connection to the portal is given in Figure 4.



```
Algorithm for student of the year:
Step 1: Fetch CGPA, number of internships and total number of certificates of all students
Step 2: Filter students on the basis of minimum criteria for qualification
Step 3: Define weightages,
        academics_weightage = 0.4
        internships_weightage = 0.3
        extra-curricular_weightage = 0.3
Step 4: Calculate score for each eligible student,
        for each student:
        {
                score= (CGPA * academics_weightage) + (no_of_internships *
        internships_weightage) + (no_of_certificates * extra-curricular_weightage)
        }
Step 5: Store all the scores of the students in the database:
        For each eligible student:
        {
                Store the score in the database
        }
Step 6: Sort all the students by their score.
Step 7: Display the top 5 students as nominees for the student of the year award.
```

Figure 3. Student of the Year Algorithm



https://unifiedsystem.tk/login

Connection is secure

Your information (for example, passwords or credit card numbers) is private when it is sent to this site. Learn more

Pop-ups and redirects: Allow (default)

Certificate (Valid)

Cookies (6 in use)

Site settings

Figure 4. SSL Secure Connection

The steps to generate certificates using Certbot on AWS EC2 Amazon Linux 2 AMI are as follows,

Prerequisite: Connection to AWS EC2 server using putty or SSH.

- In order to install and use Certbot, we need to first install the EPEL (Extra Packages for Enterprise Linux)
- Install Certbot using, sudo yum install certbot
- Generate certificate using certbot for built-in standalone webserver using,

  sudocertbotcertonly --standalone –d domainname
- Four files will be generated. We will use theprivkey.pem and fullchain.pem in our primary flask python file.

4. Results

This section provides the visuals of the system. All the different web pages of the web-application are provided along with their brief descriptions of the functions.

Figure 6 for faculty home page, which renders the unstructured data shown in Figure 5. 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 student's data. Also, the faculty will be provided with the search option along 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. Figure 7 shows the 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.

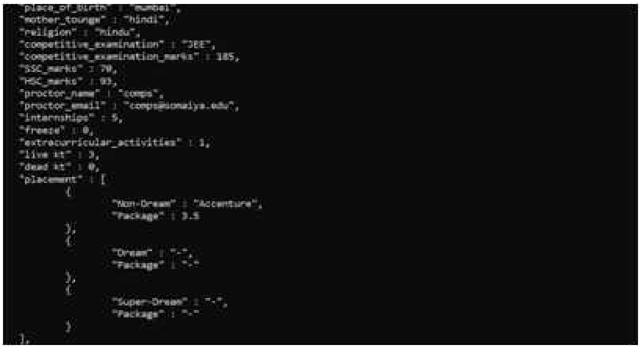
Figure 9 shows the student examination details. Figure 8 shows the unstructured format for storage of the examination details in the database. Referring to the unstructured data the data is highly nested, which is rendered to a structured format for ease of view to the user.

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 past five years. The total admissions for a particular year irrespective of the departments will be represented by a single bar graph.

Figure 10 is an example of graphical view of the students in different categories during a particular admission year. Various other types of graphs have been included in the system for data visualization.

The print table will provide the feature of exporting the current table data into a PDF document. Figure 11 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.

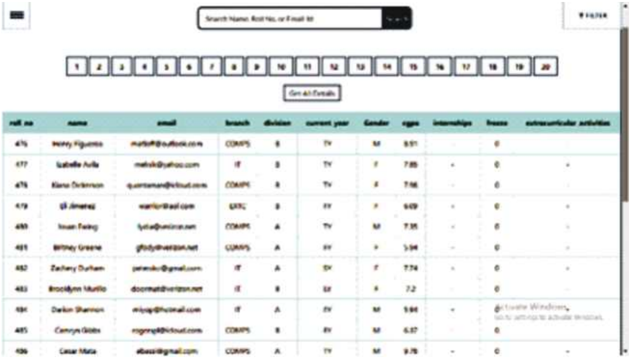


```
"place_of_birth" : "mumbai",
"mother_tounge" : "hindi",
"religion" : "hindu",
"competitive_examination" : "JEE",
"competitive_examination_marks" : 185,
"SSC_marks" : 70,
"HSC_marks" : 93,
"proctor_name" : "comps",
"proctor_email" : "comps@somaiya.edu",
"internships" : 5,
"freeze" : 0,
"extracurricular_activities" : 1,
"live kt" : 3,
"dead kt" : 0,
"placement" : [
        {
                "Non-Dream" : "Accenture",
                "Package" : 3.5
        },
        {
                "Dream" : "-",
                "Package" : "-"
        },
        {
                "Super-Dream" : "-",
                "Package" : "-"
        }
],
```

Figure 5. Storing Data in Unstructured Format

Conclusion

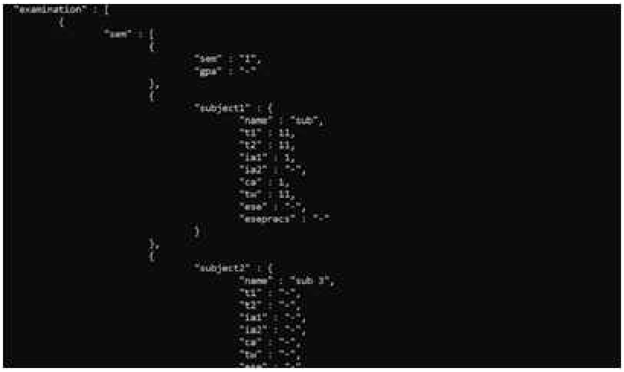
The design and the different functionalities of the ERP



roll no	name	email	branch	division	current year	Gender	cgpa	internships	freeze	extracurricular activities
476	Henry Figueroa	mattoff@outlook.com	COMPS	B	TY	M	8.51	-	0	
477	Isabelle Avila	melnik@yahoo.com	IT	B	TY	F	7.85	-	0	-
478	Kiana Dickerson	quintanar@icloud.com	COMPS	B	TY	F	7.66	-	0	-
479	Eli Jimenez	warrior@aol.com	EXTC	B	FY	F	6.69	-	0	-
480	Iman Ewing	lydia@verizon.net	COMPS	A	TY	M	7.35	-	0	-
481	Britney Greene	gfody@verizon.net	COMPS	A	FY	F	5.94	-	0	-
482	Zachary Durham	petersko@gmail.com	IT	A	SY	F	7.74	-	0	-
483	Brooklynn Murillo	docmatt@verizon.net	IT	B	SY	F	7.2	-	0	-
484	Darion Shannon	mbyag@hotmail.com	IT	A	FY	M	5.66	-		
485	Camryn Gibbs	rogerspl@icloud.com	COMPS	B	FY	M	6.37	-	0	
486	Cesar Mata	sbassi@gmail.com	COMPS	A	TY	M	9.78	-	0	-

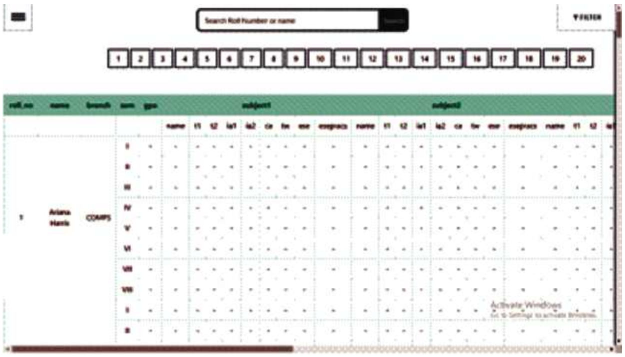
Figure 6. Faculty Home Page

Figure 7. Student Home Page



```
"examination" : [
        {
                "sem" : [
                {
                        "sem" : "I",
                        "gpa" : "-"
                },
                {
                        "subject1" : {
                                "name" : "sub",
                                "t1" : 11,
                                "t2" : 11,
                                "ia1" : 1,
                                "ia2" : "-",
                                "ca" : 1,
                                "tw" : 11,
                                "ese" : "-",
                                "esepracs" : "-"
                        }
                },
                {
                        "subject2" : {
                                "name" : "sub 3",
                                "t1" : "-",
                                "t2" : "-",
                                "ia1" : "-",
                                "ia2" : "-",
                                "ca" : "-",
                                "tw" : "-",
```

Figure 8. Student Examination Details in Unstructured Format



Search Roll Number or name
Search
FILTER
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Activate Windows
Go to Settings to activate Windows.

roll_no	name	branch	sem	gpa	subject1									subject2												
					name	t1	t2	ia1	ia2	ca	tw	ese	esepracs	name	t1	t2	ia1	ia2	ca	tw	ese	esepracs	name	t1	t2	ia
1	Ariana Harris	COMPS	I	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			II	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			III	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			IV	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			V	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			VI	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			VII	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			VIII	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			I	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			II	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Figure 9. Student Examination Details Page

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 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, a system like this can keep up with the organizational needs and will go a long way in helping the educational institutions in integrating, retrieving and maintaining their student's data.

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.

- The application can provide an end to end encrypted channel for transmission of data between the server and the client in a secured way.
- We shall incorporate data analytics into the application to get some valuable statistics about the students.

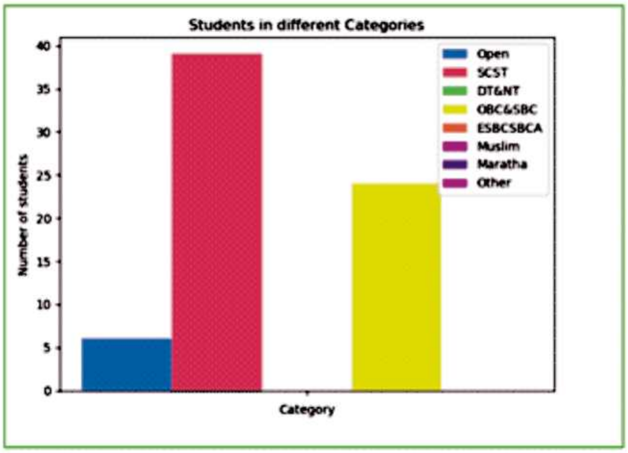
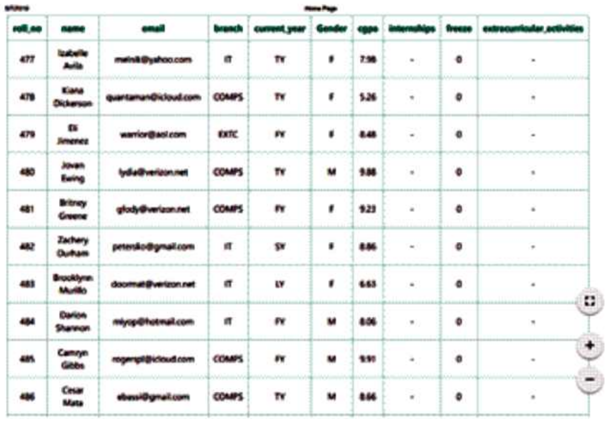


Figure 10. Plot of Students Category Wise



roll_no	name	email	branch	current_year	Gender	cgpa	internships	freeze	extracurricular_activities
477	Izabelle Avila	metnik@yahoo.com	IT	TY	F	7.98	-	0	-
478	Kiana Dickerson	quantaman@icloud.com	COMPS	TY	F	5.26	-	0	-
479	Eli Jimenez	warrior@aol.com	EXTC	FY	F	8.48	-	0	-
480	Jovan Ewing	lydia@verizon.net	COMPS	TY	M	9.88	-	0	-
481	Britney Greene	gfody@verizon.net	COMPS	FY	F	9.23	-	0	-
482	Zachery Durham	petersko@gmail.com	IT	SY	F	8.86	-	0	-
483	Brooklynn Murillo	doormat@verizon.net	IT	LY	F	6.63	-	0	-
484	Darion Shannon	mlyop@hotmail.com	IT	FY	M	8.06	-	0	-
485	Camryn Gibbs	rogerspl@icloud.com	COMPS	FY	M	9.91	-	0	-
486	Cesar Mata	ebassi@gmail.com	COMPS	TY	M	8.66	-	0	-

Figure 11. Save as pdf

References

[1]. **Babu, A., & Surendran, S. (2017).** Relational to NoSQL database migration. *International Journal of Innovative Research in Science, Engineering and Technology*, 6(5), 58-62.

[2]. **Gharanai, M. H., Gh, R. S., & Ahmadi, A. R. (2016, December).** In the digital future: Revitalizing information management systems in Afghan settings through not only SQL (MongoDB) technology. In *2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)* (pp. 45-48). IEEE.

[3]. **Hanine, M., Bendarag, A., & Boutkhoum, O. (2016).** Data migration methodology from relational to NoSQL databases. *World Academy of Science, Engineering and Technology*, *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 9(12), 2369-2373.

[4]. **Inglés, R., Perek, P., Orlikowski, M., & Napieralski, A. (2015, June).** A simple multithreaded C++ framework for high-performance data acquisition systems. In *2015 22nd International Conference Mixed Design of Integrated Circuits & Systems (MIXDES)* (pp. 153-157). IEEE.

[5]. **Lee, C. H., & Zheng, Y. L. (2015, June).** Automatic SQL-to-NoSQL schema transformation over the MySQL and HBase databases. In *2015 IEEE International Conference on Consumer Electronics-Taiwan* (pp. 426-427). IEEE.

[6]. **Liu, J., Du, X., Li, H., & Yang, J. (2017, December).** Heterogeneous Learning Resources Integration and Cross-Database Retrieval. In *2017 International Conference of Educational Innovation through Technology (EITT)* (pp. 345-346). IEEE.

[7]. **Rocha, L., Vale, F., Cirilo, E., Barbosa, D., & Mourão, F. (2015).** A framework for migrating relational datasets to NoSQL. *Procedia Computer Science*, 51, 2593-2602.

[8]. **Symeonaki, E., Papoutsidakis, M., Tseles, D., & Sigala, M. (2016, August).** Post-implementation evaluation of a university Management Information System (UMIS). In *2016 Third International Conference on Mathematics and Computers in Sciences and in Industry (MCSI)* (pp. 14-19). IEEE.

[9]. **Veen, J. S. V. D., Waaij, B. V. D., & Meijer, R. J. (2012, June).** Sensor data storage performance: SQL or NoSQL, physical or virtual. In *2012 IEEE Fifth International Conference on Cloud Computing* (pp. 431-438). IEEE.

[10]. **Zhao, G., Lin, Q., Li, L., & Li, Z. (2014, November).** Schema conversion model of SQL database to NoSQL. In *2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 355-362). IEEE.

ABOUT THE AUTHORS

Nishith Khandor is working as a Consultant in Cyber Risk Advisory at Deloitte Touche Tohmatsu India. He pursued B.Tech in Information Technology from K.J. Somaiya College of Engineering, Vidyavihar in the academic year 2018 - 2019. He has attended sessions regarding network and security. He has been part of many competitive programming competitions. Nishith Khandor holds a certificate for Java course offered by Oracle Academy and Microsoft Technology Associate for Security.

Dhruv Mehta is currently pursuing his Masters of Science in Computer Science from the University of Texas at Dallas. He completed his B.Tech in Information Technology from K.J Somaiya College of Engineering in 2019. He is a certified Ethical Hacker and has attended various seminars/workshops on Competitive Coding, Java Programming, Blockchain and Cyber Security. He is currently a member of EC Council and was a member of Computer Society of India.

Zirak Mistry is working as a Graduate Engineering Trainee (GET) in the Information Technology subsidiary of the Larson and Toubro Company Ltd. He is a B.Tech Graduate in Information Technology. He completed his graduation from the K.J. Somaiya College of Engineering, Vidyavihar in the academic year 2018 - 2019. He has participated in conferences regarding Cloud-Computing and Competitive Programming. He is a member of the Computer Society of India (CSI) since 2015. Zirak Mistry is the part of the Oracle academy and has completed the online course on core Java provided by Oracle.

Ayush Mittal is working as an Analyst at Ernst and Young. He is a B.Tech Graduate in Information Technology. He completed his graduation from the K.J. Somaiya College of Engineering, Vidyavihar in the academic year 2018 - 2019. He has participated in Conferences regarding Cloud Computing and Big Data. He is a member of the Computer Society of India (CSI) since 2016.

Sagar Korde is working as an Assistant Professor in the Department of Information Technology at K. J. Somaiya College of Engineering, Mumbai, India. He holds a Master of Engineering Degree with a specialization in Embedded System and Computing. He has published a handful of papers in the same areas. His areas of interest and work include Embedded System, Internet of Things, Computer Vision, Operating System, and Usability Engineering.