A Robust Plagiarism Checker for Detecting Copy Violations Documents and Images

[1]Douglas, E.T, [2]Ansa, G.O, [3]Frank, J.W. & [4]Longe, O.B
[123]Department of Computer Science, Akwa Ibom State University, Ikot Akpaden, Akwa Ibom State, Nigeria
[4]Department of Information Systems, American University of Nigeria, Yola, Nigeria
E-mail(s): udyted1993@gmail.com; godwinansa@aksu.edu.ng; mimiyoung2004@gmail.com; Olumide.longe@aun.edu.ng
Phones: +2347068138009; +2348109218888; +2348064457448

ABSTRACT

We developed a robust Plagiarism checker for detecting documents and images using Computer Automated External detection technique in order to accept all file format, determine plagiarized documents and images. There are two levels of search, which is the online and offline. The online search level requires the Google search API to analyze documents. This analysis is done using the Representational State Transfer (REST) technology which provides interoperability between computer systems on the Internet. The offline search works with Rabin-Karp algorithm to compare two textual digital documents for matches using hash function which determines the percentage by converting each string into numbers called hash value. The Image Pixel Analysis Method (IPAM) hypothesis compares two images by analyzing individual pixels of the two images being compared within any document. Then a percentage similarity score for both search level is generated. This approach is being evaluated against the intrinsic detection technique and the results are very promising.

Keywords: Referencing, Document, Plagiarism, Hash function, String-matching.

iSTEAMS Proceedings Reference Format
Douglas, E.T, Ansa, G.O, Frank, J.W. & Longe, O.B (2019): A Robust Plagiarism Checker for Detecting Copy Violations Documents and Images. Proceedings of the 17th iSTEAMS Multidisciplinary Research Nexus Conference, D.S. Adegbenro ICT Polytechnic, Itori-Ewekoro, Ogun State, Nigeria, 21st – 23rd July, 2019. Pp 29-40. www.isteams.net - DOI Affix - https://doi.org/ 10.22624/AIMS/iSTEAMS-2019/V17N1P4

1. BACKGROUND TO THE STUDY

Plagiarism is an act of fraud. It involves both stealing someone else's work and lying about it afterward. The effect of this phenomenon has been even more pronounced following the wide spread use of the Internet and digital documents which are easily copied. Students who plagiarize their reports, essays or programming assignments are usually void of understanding of the concepts described in the course work and are therefore not regarded to have undergone the proper learning process. Plagiarism in the sense is "theft of intellectual property" Plagiarism is derived from the Latin word "plagiarius" which means kidnapper. It is defined as "the passing off of another person's work as if it were one's own, by claiming credit for something that was actually done by someone else". In the last few decades, it was a challenge to check the similarity between two documents. This challenge is what triggers research efforts to provide practical approach for detecting plagiarism. Digital documents are easily copied due to the nature of the documents themselves. In academic evaluations, students often copy each other's assignments thereby undermining the purpose of teaching which is to pass on instruction and knowledge to the students. Many software tools exist for checking and assisting in monotonous and time consuming task of tracing plagiarism Hoad & Zobel. (2003). Identifying the owner a whole text document is practically difficult and impossible for markers. Examples of tools used to detect plagiarized works are Plagiarism, plagium, PlagTracker, KatchPlaiger etc. KatchPlaiger uses string pattern matching to determine similarities between two textual digital documents. The software generates a similarity score which may be represented as a percentage that indicates the degree of similarity that exists between two digital documents.

1.1 Statement of Problem

Consequently, there is a need to develop a plagiarism checker for ascertaining the originality of student's project to address these problems in tertiary institutions.

- Plagiarism discourages creativity, invention or innovation.
- It deprives students of the intended learning objectives of a paper by not conducting research and formulating an original document.
- The student lacks the ability of mastering good writing skills like research, citing sources and structuring an essay.
- Relying solely on other people's work keeps students from crafting their own voices as writers and developing them throughout their educational careers.

1.2 Aim and Objectives of the Research

The aim of this project is to develop a more practicable and reliable solution that can easily identify plagiarism in images and multiple documents thereby improving the overall chances of detecting plagiarism.
The objectives of this research work include:

- The proposed plagiarism software will generate a similarity score which may be represented in percentage that indicates the degree of similarity that exists between two digital documents.
- Check for plagiarized projects and detect plagiarized images.
- This system supports different document format including PDF, TXT, RTF, and DOCX using HTML, JavaScript, PHP, and MySQL Database Management System.

2. RELATED WORKS

The Intrinsic Plagiarism Detection in Digital Data (IPDDD) was proposed by (Netra et a., 2015) to allow the examiner of the research papers or the editor of digital journals to determine whether there are plagiarized sentences in the submitted research paper in text format only. IPDDD basically attempts to detect plagiarized sentences in the digital text data without using a reference corpus. IPDDD uses grammar analysis of the sentences written by the author. This system detects plagiarism in papers only available in .txt format. **Ranti, Andysah, & Utama, (2017)** adopted the Rabin-Karp algorithm for Examination of Document Similarity that searches for a substring pattern in a text using hashing. It is very effective for multi-pattern matching words (Sharma J. & Singh M., 2015). One of the practical applications of Rabin-Karp's algorithm is plagiarism detection. Rabin-Karp relies on a hash function to determine the percentage of plagiarism. The hash function is a function that determines the feature value of a particular syllable fraction. It converts each string into a number, called a hash value. Rabin-Karp algorithm determines hash value based on the same word. Rabin-Karp requires a large prime number to avoid possible hash values similar to different words. The disadvantage of this algorithm is that the system can never know which documents came first. The algorithm can only determine the similarities that occur in the comparable documents but does not calculate match scores between a query and documents which are sorted decreasingly by their scores, and highly ranked documents are then returned. **Lefteris, M. & Athena, V. (2005).** The PDetect functionality consists of the source code representation, the definition of the similarity measure and the clustering algorithm that detects the clusters of plagiarism. PDetect operates in two phases. Phase 1 processing extracts the pairwise similarities from a given set of programs and then in phase 2 processing the pairwise similarities are given as input and the clusters of plagiarism are detected.

Mena, M. (2012) propose a plagiarism detection tool for Arabic documents (Aplag). Aplag is based on heuristics to compare suspected documents at different hierarchical levels to avoid unnecessary comparisons. In addition, to address the problem of rewording, Aplag replaces each word's root by the most frequent synonym extracted from Arabic WordNet. **Longe & Kolawole (2012)** addressed the issue of dliberate or inadvertent replication of digital documents occasioned by the volume of digital resources made available on the world wide-web and the ease with which they can be copied (plagiarized) without degradation in quality and content and posited that the menace has emerged as one unintended consequence of the internet. Using students' programming assignment as test data, they designed and implemented a system tagged "KatchPlaiger" using standard Object Oriented design approach. Coding was done using the Java Programming language. The system targeted at providing a University-wide solution to the problem of unauthorized duplication of digital contents employ string pattern matching to determine similarities between two textual digital documents. The software implements the Running-Karp-Rabin Greedy String Tilling algorithm (RKR-GST) and generates a similarity score which is represented as a percentage that is indicative of the degree of similarity that exists between digital documents. Visual representations are provided to aid the understanding of the system output**.**

Alzahrani, S., (2015) system goes through four main steps: (i) Pre-processing which includes tokenization and stop-word removal, (ii) Retrieve a list of candidate source documents for each suspicious document using n-gram fingerprinting and Jaccard coefficient, (iii) An in-depth comparison between the suspicious documents and the associated source candidate documents using *k*-overlapping approach (iv) Post-processing where consecutive *n*-grams are joined to form united plagiarized segments. **Magooda et al., (2015)** propose an extrinsic plagiarism detection system named RDI_RED. In this system, Lucene search engine is used to select a list of candidate source documents. The candidate documents are aligned to detect plagiarized segments (aligned parts). Finally, a set of rules is applied by a filtering module in order to filter the aligned parts. RDI_RED system can be easily deployed on-line. Though, it does not address synonyms substitution and paraphrasing.

3. ANALYSIS OF THE EXISTING SYSTEM

The existing system is built around the Computer Automated Intrinsic Syntax Similarity Based Detection. This methodology aims at detecting Paraphrasing, Idea, Mosaic and 404 Error textual types of plagiarisms that may possibly be observed in the submitted research paper. The Intrinsic Plagiarism Detection in Digital Data (IPDDD) system allows the examiner of the research papers or the editor of digital journals to determine whether there are plagiarized sentences in the submitted research paper in text format only. IPDDD basically attempts to detect plagiarized sentences in the digital text data without using a reference corpus. IPDDD uses grammar analysis of the sentences written by the author. If suspicious sentences are found by computing the similarity distance between grammar trees of the sentences found in the digital data source to that of the successive sentences, then by calculating appropriate mathematical values using the computed distances between pairs of grammar trees and a certain threshold value, the software tries to identify suspicious sentences. Then such sentences are recorded and their total count is stored. Using the count of plagiarized sections and the total number of sentences in the paper, an authenticity ratio is calculated. If the percentage ratio is more than a prescribed value, then the paper is decided to be violating the rules of plagiarism acceptance. If suspicious sentences are found by computing the similarity distance between grammar trees and computing mathematical parameters by comparing the edit distances with the mean value, the software declares them as potentially plagiarized sentences.

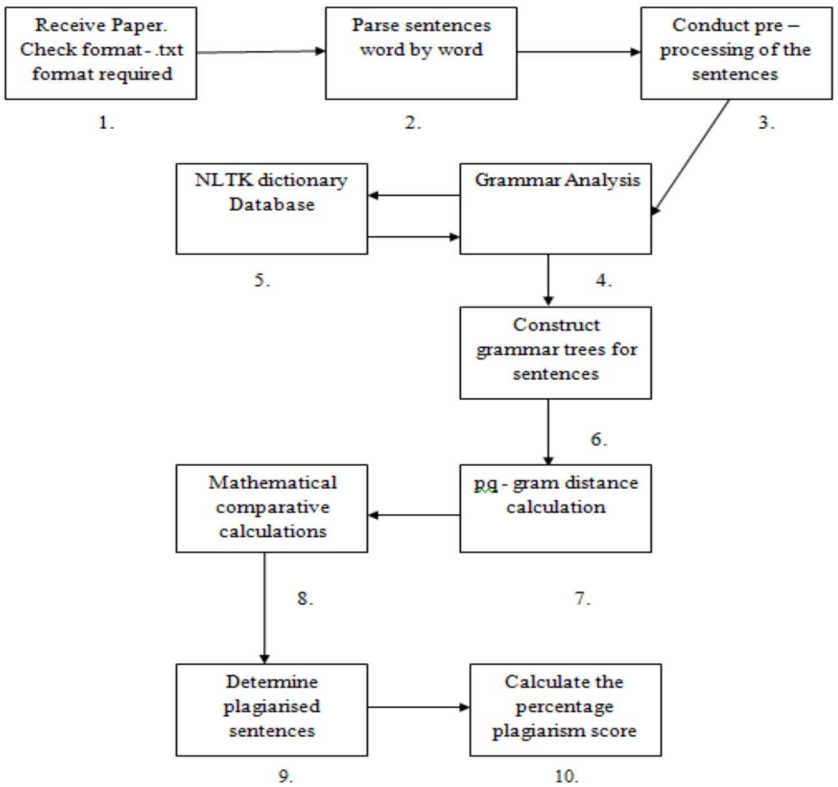

Fig. 5.1 Existing Architecture of IPDDD

4. PROPOSED SYSTEM ARCHITECTURE

In order to detect different types of plagiarism, our proposed system is based on Computer Automated External detection technique. This system receives and determines documents in (PDF, RTF, DOC, DOCX, and TXT) formats. All sentences in the digital data document are parsed individually to undergo pre-processing. Copies of the sentences are made and is hashed and stored in English. When digital document are parsed, the search is carried out on online using the Google search API to analyze documents. This analysis is done using the Representational State Transfer (REST) technology which provides interoperability between computer systems on the Internet. Rabin-Karp algorithm to compare two textual digital documents for matches using hash function which determines the percentage by converting each string into numbers called hash value. The Image Pixel Analysis Method (IPAM) hypothesis compares two images by analyzing individual pixels of the two images being compared within any document. To determine the percentage score of various types of plagiarism, the mathematical comparative calculates the median values for every row in the distance matrix and store these values in a list and the plagiarized documents stored in the list is sorted decreasingly by their scores. This generates the percentage similarity score.

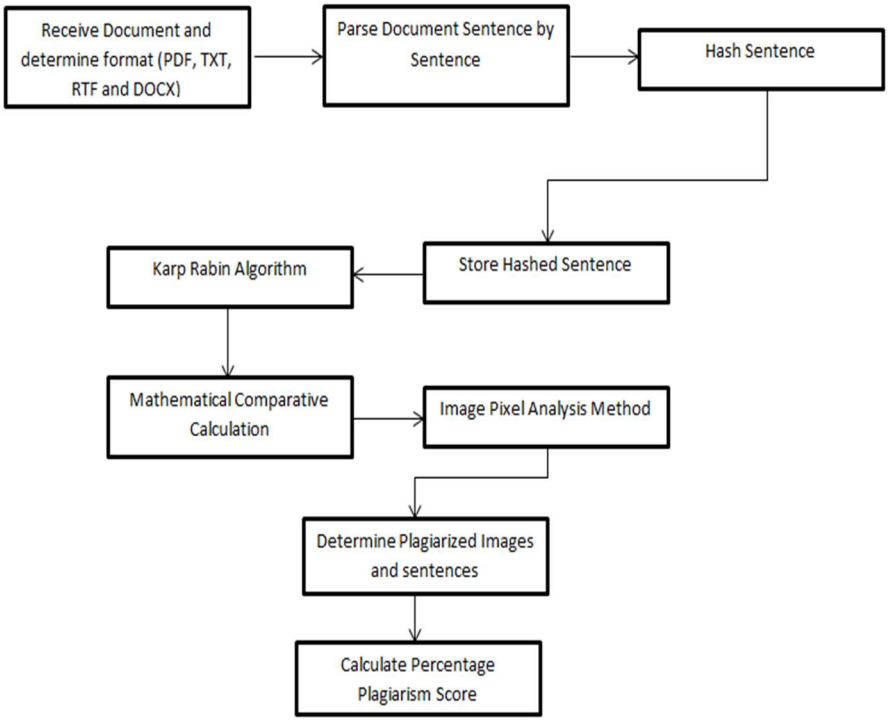

Fig. 6.1 Proposed System Architecture

5. RESULTS AND DISCUSSION

Rabin- Karp Calculation

Hashing is the most important step in the Rabin-Karp algorithm. The result of hashing letters of k-gram with a certain number of bases is obtained by multiplying the American Standard Code for Information Interchange (ASCII) value with predetermined numbers where the base is prime. Rabin-Karp method has provisions if two strings are same then the hash value must be the same as well. Assume the text is MEDAN

K-GRAM = 5

BASIS = 7

A = MEDAN

A (1) = 77
A (2) = 69
A (3) = 68
A (4) = 65
A (5) = 78
Hash = $(77 * 7^4) + (69 * 7^3) + (68 * 7^3) + (65 * 7^2) + (78 * 7^1)$
=235599

Table 1: Hash Value of Document A

19875	~~16830~~	23124	17433	20546
21489	~~26753~~	13498	23846	16528
21848	28447	29994	10301	~~13009~~
18832	27217	23157	25854	~~22492~~
14952	14337	29348	19978	28809
13485	14188	~~13131~~	~~21215~~	12053
~~25669~~	13809	26508	19455	25356
29964	17723	26633	17445	11803
19477	27142	24814	15155	26266
~~28432~~	19007	~~21896~~	16625	~~20681~~

Table 2: Hash Value of Document B

~~28432~~	26406	28424	13930	19187
18049	10867	18516	~~26753~~	19975
10152	13053	24120	~~21896~~	18351
12605	25101	~~21215~~	20750	15513
22949	26006	25045	25932	10695
13254	21504	20286	~~22492~~	10615
25565	29941	17403	23018	22666
19744	19769	19877	29535	13139
~~25669~~	~~16830~~	14297	20916	24640
16960	~~20681~~	~~13131~~	~~13009~~	18947

There are ten pieces of the same hash that both tables have. Then, after calculating the similar hash value, to calculate the percentage of similarity of the two documents. The formula used is as follows:

$$P = \frac{2 * SH}{THA + THB} * 100\%$$

Where:
P = Plagiarism Rate
SH = Identical Hash
THA = Total Hash in Document A
THB = Total Hash in Document B

$$P = \frac{2*10}{50+50} * 100\%$$

$$= \frac{20}{100} * 100\%$$
$$= 0.5 * 100\%$$
$$= 20\%$$

The percentage of plagiarism held by both documents is 20%

5.1 Mathematical Analysis to Determine Plagiarized sentences

When the distance between two strings which is the source string (S) and the target string (T) throughout the document is obtained, we must find out the possibly plagiarized sentences. First of all we calculate the median values for every row in the distance matrix and store these values in a list. Then, we compute the mean of all the values in the median list using the below formula:

$$\sum \frac{f(K-gram) \times S}{N}$$

Where: "K-gram" represents the total number of characters in a sentence.
"S" represents the total similarity score of a particular length of string or sentence.
"N" represents the total number of partitioned sentences.

5.2 Detecting Image Plagiarism in Digital Documents

Image plagiarism in this context narrates the unattributed use of images and graphical illustrations in documents. The first idea that comes to mind is using a "One Way Function" like the MD5 hash. Here MD5 hashes of two graphic objects are directly compared to check for equality. One obvious flaw with using only One Way Hash Functions like MD5, SHA1 or SHAII is that any slight variation (even as small as a dot) will be flagged as disparity. As such the images will be considered as being different allowing the plagiarized image to pass the plagiarism test. For example let's say User A copies any image or graphical illustrations from the work of (projects or research of any kind) of User B without attribution.

All user needs to do is to convince this method (one way function) that the two images are different by adding a little dash or dot or any character at all. This will make the hashes of image A different from that of image B, therefore will comfortably pass the test. However, this should not be the case; any slight variations should not raise a false alarm. The Image Pixel Analysis Method (IPAM) hypothesis compares two images by analyzing individual pixels of the two images being compared. This method defeats any attempt to fool the system by making slight modifications to the images.

By modification the following are taken into consideration:

- Changing the images to grey scale.
- Changing the image to black and white.
- Changing the orientation of the image.
- Changing the size by scaling.

10 × 9 pixels

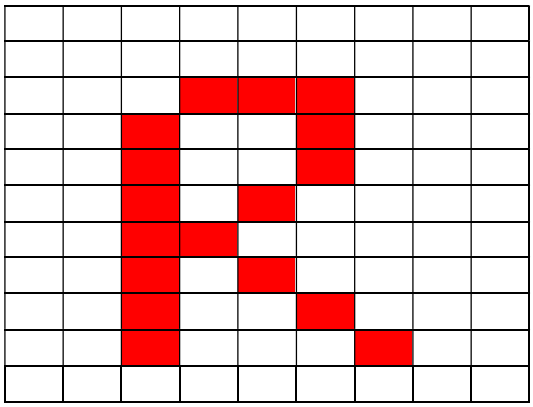

Image A

Plate 1

10 × 9 pixels

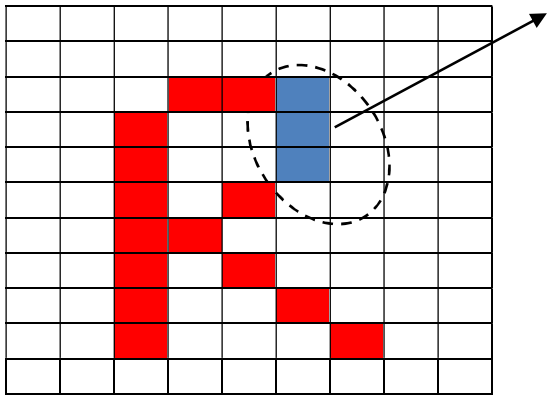

Image B

Plate 2

6. IMPLEMENTATION

This software employs tools like bootstrap and jquery library which makes it easier for the front end to interact with the document object elements and handle events when a file format is received. The program uses Representational State Transfer (REST*)* to carry out analysis between two document online and Rabin Karp algorithm to compare two textual documents. It requires a chrome browser and a JavaScript engine to run analysis to check for plagiarized work. This software also checks for image plagiarism within a document using the Image Pixel Analyzes Method (IPAM). The results are displayed and percentage score is shown. With this score it can be decided if a student should review a project for being rated high for plagiarism or not. Then results can be printed out in PDF format if need be.

- **Software Interface:** This shows the software interface which includes where the students will register and login as well as the lecturer.

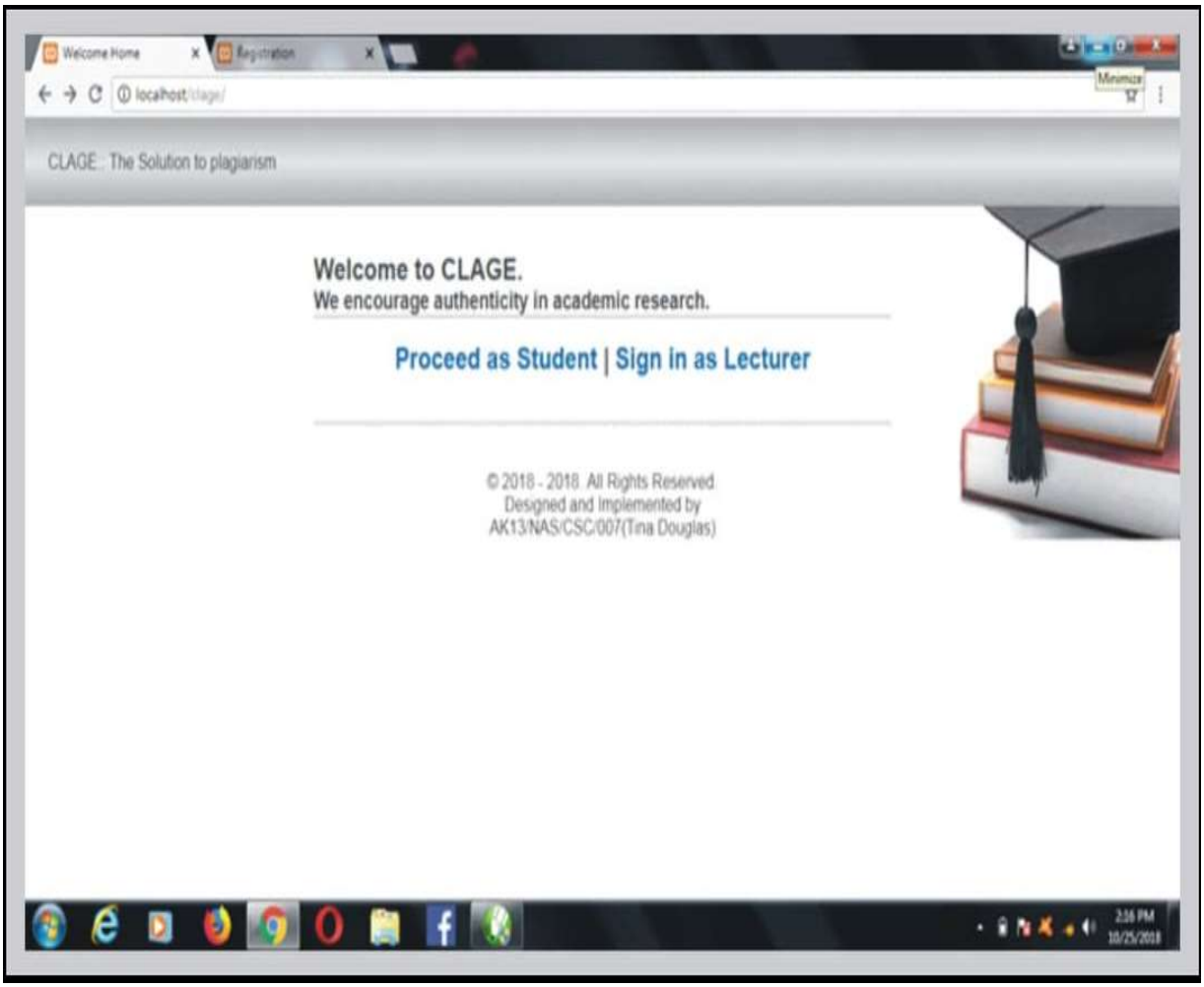

Fig. 1: Software Interface

- **Project Upload:** The fig below enables the student to upload documents. Once the document is uploaded, it dictates the file format, matriculation number and the title of the document.

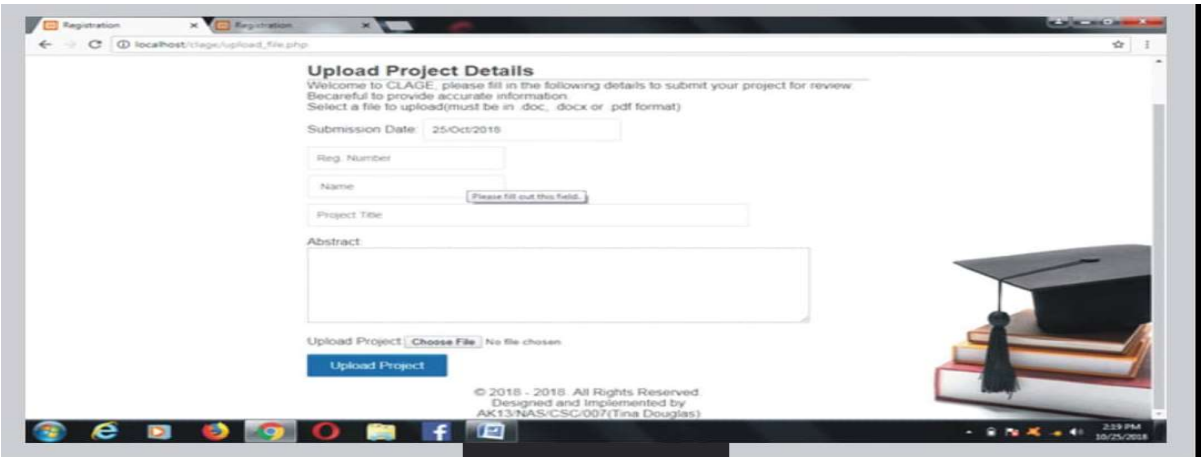

Fig. 2: Project Upload

- **Percentage Plagiarism Score**: This displays the percentage similarity score of the two textual documents for plagiarism, if rated above 50% the student is asked to review the work.

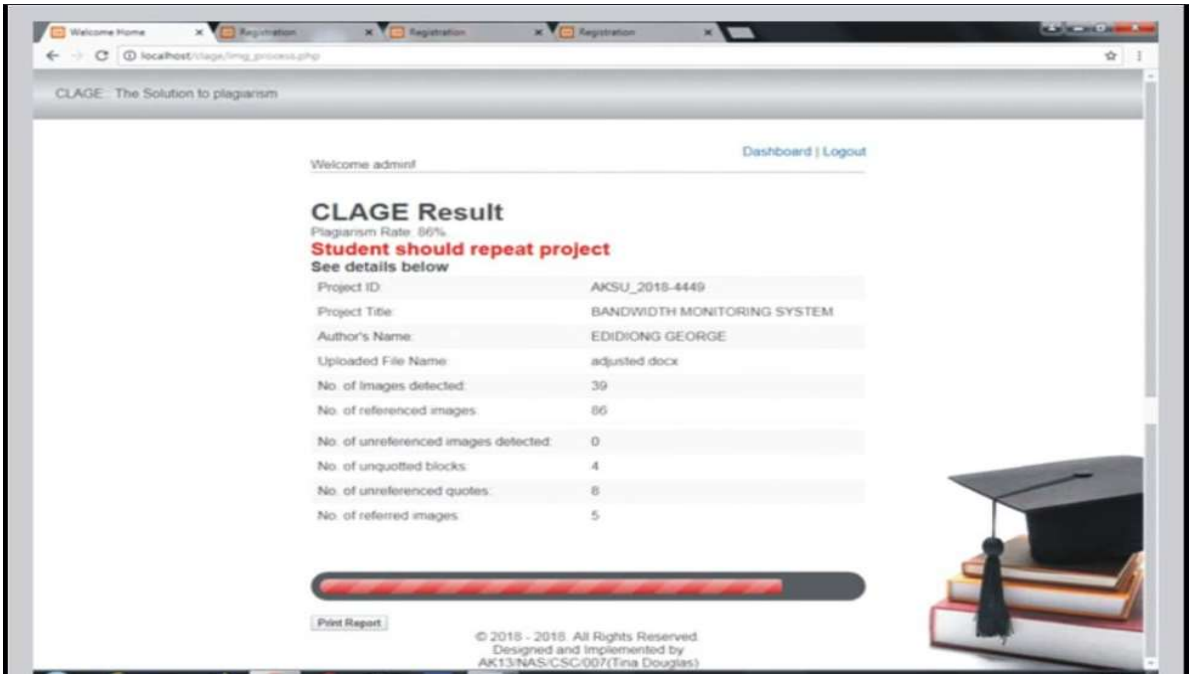

Fig. 3: Percentage Similarity Score

- **Database of Uploaded Projects:** Here all document compared online for plagiarism is stored in a database for subsequent retrieval.

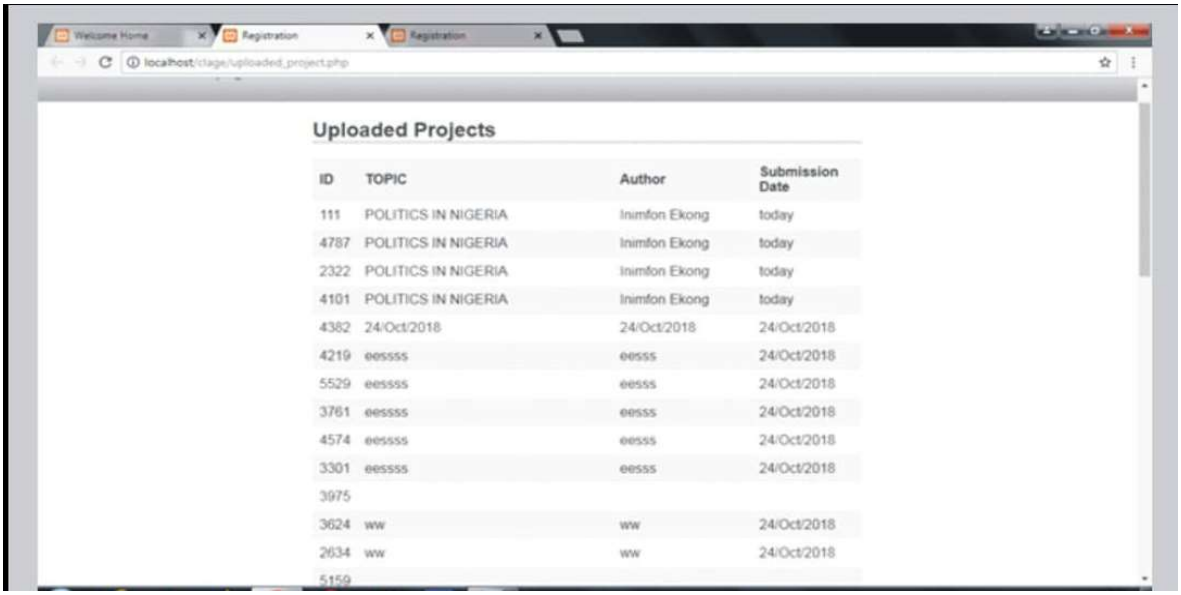

Fig..4: Database for uploaded Projects

7. CONCLUSION

The proposed system CLAGE – plagiarism checker for detecting documents and images adopted the Computer Automated External Detection Technique to improve on the intrinsic detection technique used in the existing system for purpose of the objectives of this study. This system supports different file formats like. txt, docx, rtf and pdf. The system also operates on two levels: using online and offline searches. The retrieval of source documents on the Web is achieved using Google API and Representational State Transfer (REST) which provide interoperability between the computer system and the Internet. CLAGE is able to compare two textual digital documents using the Rabin-Karp algorithm. The comparative mathematical calculation locates plagiarized sentences stored in the list by their score. CLAGE compares images within a document using IPAM. The limitation of this work is that images within PDF document cannot be compared because it compresses individual pixel. This reduces the possibility of checking for similarity between two images within a pdf document.

8. FUTURE SCOPE

This project is aimed at detecting plagiarized documents and images available in .txt, .docx, .rtf and .pdf formats. The results show that images with PDF documents cannot be compared. As future work, this system (CLAGE) will be extended to cover the detection of image plagiarism in PDF documents. The optimization of the mathematical parameters to increase the accuracy and speed of plagiarism detection will also be further improved in the extended version of CLAGE.

REFERENCES

1. Alzahrani, S. (2015).*Arabic Plagiarism Detection Using Word Correlation in n-Grams with k-Overlapping Approach*. Working Notes for Panaraplagdet: pp. 123-125.
2. Hoad, T., & Justin, Z. (2003). *Methods for Identifying Versioned and Plagiarized Documents*. Journal of the American Society for Information Science and Technology: 54(3):203–215.
3. Lefteris, M. & Athena, V. (2005*). PDetect: A Clustering Approach for Detecting Plagiarism in Source Code Datasets*. Published by Oxford University Press. Greece.
4. Longe, O.B & Kolawole, O. (2012). Running-Karp-Rabin Greedy String Tilling Algorithm (RKR-GST) based System for Determining Similarities between Textual Digital Documents. 4th International Conference on ICT for Africa, Makerere University, March, 2012.
5. Menai, M. (2012). *Detection of Plagiarism in Arabic Documents*. International Journal of Information Technology and Computer Science (IJITCS), Vol. 4, No 10, p. 80.
6. Magooda, A., Mahgoub, A., Rashwan, M., Fayek, M. & Raafat, H. (2015) Rdi *System for Extrinsic Plagiarism Detection (Rdi Red)*. Working Notes for Panaraplagdet: FIRE Workshops pp. 126-128.
7. Netra, C., Kushagra, D., Smit, B. & Radha, S. (2015). *Intrinsic Plagiarism Detection in Digital Data*. Information
8. Technology Dept., Sardar Patel Institute of Technology, Andheri (W), Mumbai, India International Journal of Innovative
9. and Emerging Research in Engineering Volume 2, Issue 3.
10. Ranti, P., Andysah, P. & Utama, S. (2017). *Examination of Document Similarity Using Rabin-Karp Algorithm.*
11. Sharma, J. & Singh M. (2015). *CUDA based Rabin-Karp Pattern Matching for Deep Packet Inspection on a Multicore GPU*. International Journal of Computer Network and Information Security.