International Journal of Innovative Research in Computerand Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

||Volume 8, Issue 6, June 2020||

Online Plagiarism Checker Using Text Mining

Zubeda Khan, Harmeet Singh Jadhav, Ibrahim Khan, Aakash Ganjave

Assistant Professor, Dept. of Computer Science, KC College of Engineering, Thane, India

Final Year Student, Dept. of Computer Science, KC College of Engineering, Thane, India

Final Year Student, Dept. of Computer Science, KC College of Engineering, Thane, India

Final Year Student, Dept. of Computer Science, KC College of Engineering, Thane, India

ABSTRACT: Plagiarism is a major problem for research. There are, however, divergent views on how to define plagiarism and on what makes plagiarism reprehensible. In this paper we explicate the concept of "plagiarism" and discuss our project for checking plagiarism in any document. Our project is based on removing plagiarized text in the assignments given by the students using decision tree algorithm based on text mining. We first place our assignment in the space given then we click on the button named “Check for Plagiarism”. The programs run behind and we get the results. There we can view how much text is plagiarized and how much is not plagiarized. The best feature of our project is we can even check that the content is copied from the internet or not by just pasting the link of references inside the document.

KEYWORDS: Plagiarism,NET Framework Class Library,Active Data Objects, MySQL.

I. INTRODUCTION

Plagiarism is defined as the use or close imitation of the language and thoughts of another author and the representation of them as one's own original work. Plagiarism comes from a Latin verb that means, “To kidnap” If we plagiarize it means that we are kidnapping and stealing others hard work and intellectual property, which is a form of academic and public dishonesty. By the use of synonyms, plagiarism can be done. Therefore, they are difficult to recognize by the commercial software. Plagiarism affects the education quality of the students and thereby reduce the economic status of the country. Plagiarism is done by paraphrased works and the similarities between keywords and verbatim overlaps, change of sentences from one form to another form, which could be identified using WordNet etc. Academics know that student valuable learning experience is supported with the help of information, but by the use of plagiarism these experiences is demolished. Regarding project-based activities for academics it is believed that plagiarism cannot be done easily but still some students try to plagiarize by copying the work done by the other students, which is difficult for the faculty to find out. Juan et al. created a tool called beagle, which uses some collusion method to identify plagiarism. This software measures the similar text that matches and detects plagiarism. Internet has changed the student’s life and also has changed their learning style. It allows the student to deeper the approach towards learning and making their task easier. Many methods are employed in detecting plagiarism. Usually plagiarism is done using text mining method.

Scope and Objective

Plagiarism detection is of special interest to educational institutions, and with the proliferation of digital documents on the Web the use of computational systems for such a task has become important. While traditional methods for automatic detection of plagiarism compute the similarity measures on a document-to-document basis, this is not always possible since the potential source documents are not always available. We do text mining, exploring the use of words as a linguistic feature for analysing a document by modelling the writing style present in it. The main goal is to discover deviations in the style, looking for segments of the document that could have been written by another person. This can be considered as a classification problem using self-based information where paragraphs with significant deviations in style are treated as outliers. This so-called intrinsic plagiarism detection approach does not need comparison against possible sources at all, and our model relies only on the use of words, so it is not language specific. We demonstrate that this feature shows promise in this area, achieving reasonable results compared to benchmark models.

Existing methods can be classified in to two categories extrinsic and intrinsic. Extrinsic methods basically include comparison of suspicious document with the genuine works. For that several comparisons, methods have been suggested. Intrinsic methods they are opposed to extrinsic one, they examine features like linguistic one without doing comparison with the external documents. It basically aims at lexical features, syntactic features-word frequencies, structural features like average length of the paragraph.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

||Volume 8, Issue 6, June 2020||

II. PROPOSED ALGORITHM

The Project is loaded in Visual Studio 2010. We used Visual Studio for Design and coding of project. Created and maintained all databases into SQL Server 2008, in that we create tables, write query for store data or record of project.

Considering the anomalies in the existing system computerization of the whole activity is being suggested after initial analysis. The web application is developed using Asp.net with C# and SQL Server. User itself accesses this system.
User can access the web portal by logging in with valid login credentials. User need to upload a document, which will be scanned by the system and divided in 2 parts as content and links. All the filtered link will be stored separately using which system will visit each link and scan the content of that webpage. Content of the original document will be matched word to word with site's content and bring matching no. of sentences. Sentences that has matched will keep incrementing the counter and those matched content will be ignored while scanning in other sites. System will bring other websites from that website until 4 levels and perform the same if any content is matched in matched in previous level. The uploaded document is also checked with the previously stored students' documents. Final output of the system will be provided in percentage (%) form of content copied from and list of sites and matched percentage. User can view their previous history of uploaded documents. Web application allows user to view their profile and change password whenever required.

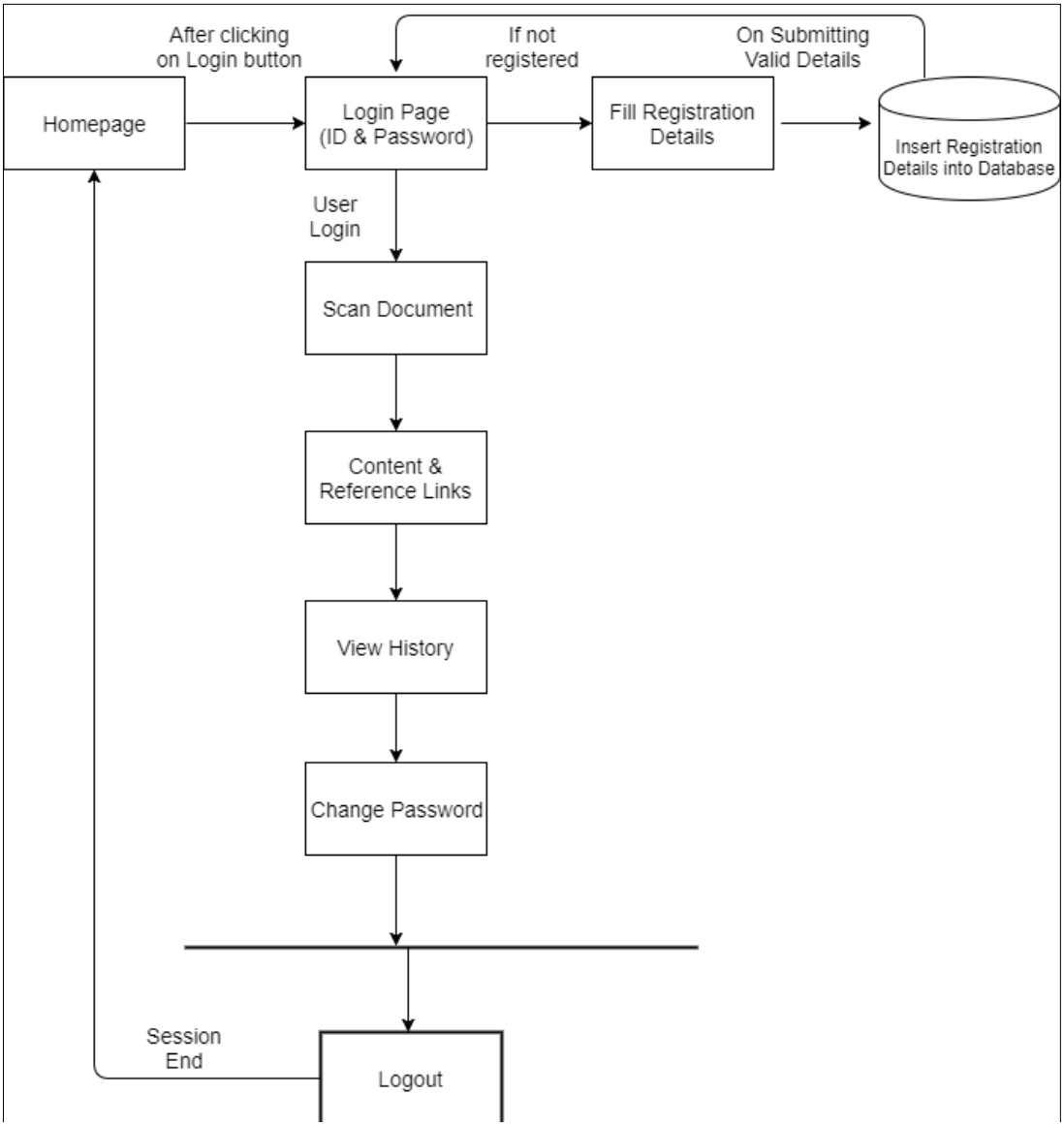

Fig.2.1.Flow Chart

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

||Volume 8, Issue 6, June 2020||

The system comprises of 2 Entities with their sub-modules as follows:

- **User:**
  1. **Registration:** User can register with their basic registration detail and create a valid login id and password.
  2. **Login:** User can login into their personal account using valid login credentials.
  3. **Upload/Scan Document:** User can upload a document file (.doc), which will be divided in 2 parts.
     - Content
     - Reference Links
     - Web application will process the content, visit each reference link, and scan the content of that webpage to match the original content.
     - Plagiarism is also checked with the uploaded documents by the students.
  4. **View History:** User can view history of their previous documents.
  5. **Change Password:** User can view their profile and change password whenever required.

- **Faculty:**
  1. **Login:** Faculty person can login using valid credentials in order to access the system.

Add Topic: New topic can be added by the faculty itself and set a due for that particular topic. The topic can be assigned to single or multiple students.

2. **Approve Student:** Faculty can view list of registered new student approve individually.
3. **View Uploads:** All the documents uploaded by the student can be viewed by the faculty person.

III. METHODOLOGY

Active Data Objects

ADO.NET is an evolution of the ADO data access model that directly addresses user requirements for developing scalable applications. It was designed specifically for the web with scalability, statelessness, and XML in mind. ADO.NET uses some ADO objects, such as the Connection and Command objects, and also introduces new objects. Key new ADO.NET objects include the Dataset, Data Reader, and Data Adapter.

The important distinction between this evolved stage of ADO.NET and previous data architectures is that there exists an object -- the Dataset -- that is separate and distinct from any data stores. Because of that, the Dataset functions as a standalone entity. You can think of the Dataset as an always disconnected record set that knows nothing about the source or destination of the data it contains. Inside a Dataset, much like in a database, there are tables, columns, relationships, constraints, views, and so forth.

A Data Adapter is the object that connects to the database to fill the Dataset. Then, it connects back to the database to update the data there, based on operations performed while the Dataset held the data. In the past, data processing has been primarily connection-based. Now, in an effort to make multi-tiered apps more efficient, data processing is turning to a message-based approach that revolves around chunks of information. At the Centre of this approach is the Data Adapter, which provides a bridge to retrieve and save data between a Dataset and its source data store. It accomplishes this by means of requests to the appropriate SQL commands made against the data store.

The following sections will introduce you to some objects that have evolved, and some that are new. These objects are:

- **Connections.** For connection to and managing transactions against a database.
- **Commands.** For issuing SQL commands against a database.
- **Data Readers.** For reading a forward-only stream of data records from a SQL Server data source.
- **Datasets.** For storing, removing and programming against flat data, XML data and relational data.
- **Data Adapters.** For pushing data into a Dataset, and reconciling data against a database.

.NET Framework Class Library

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive

functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

Decision Tree Algorithm in Text Mining

An important goal of text mining is to sift through large volumes of text to extract patterns and models that can then be incorporated in intelligent applications, such as automatic text categorizers and routers. Decision rules and decision tree-based approaches to learning from text are particularly appealing, since rules and trees provide explanatory insight to end-users and text application developers. Our research has focused on applying these methods to automatic text categorization, and more recently, developing methodologies for maximizing their performance. Our initial methodology for automatic text categorization was built around the use of rule induction, coupled with a new approach to constructing feature vectors, that emphasized the use of local dictionaries and numerical features. This helped us in achieving the end goal of defining the maximum accurate plagiarized percentage of a document.

IV. DATA FLOW DIAGRAM

A data flow diagram is graphical tool used to describe and analyse movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gain and Sarson notation develops the data flow diagrams. Each component in a DFD is labelled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD's is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a "bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

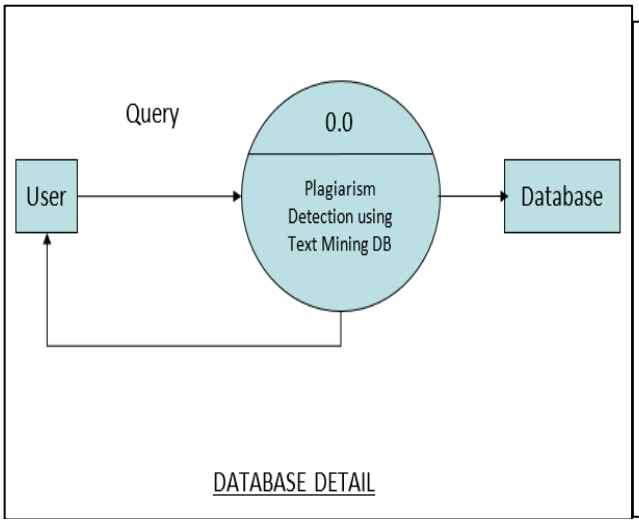

Fig.4.1 DFD Level 0

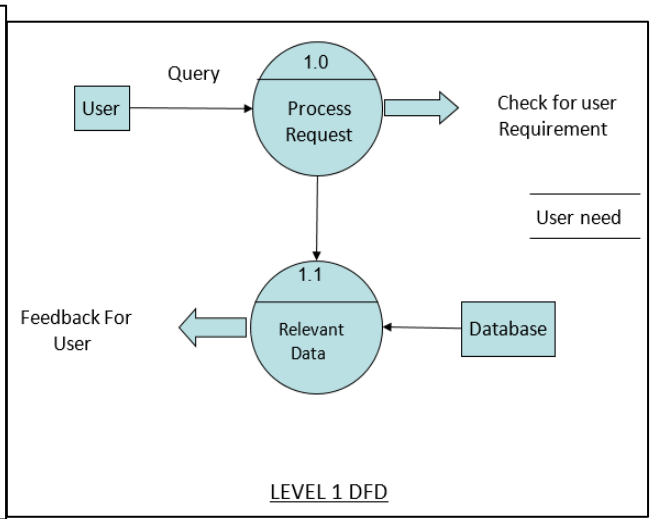

Fig.4.2 DFD Level 1

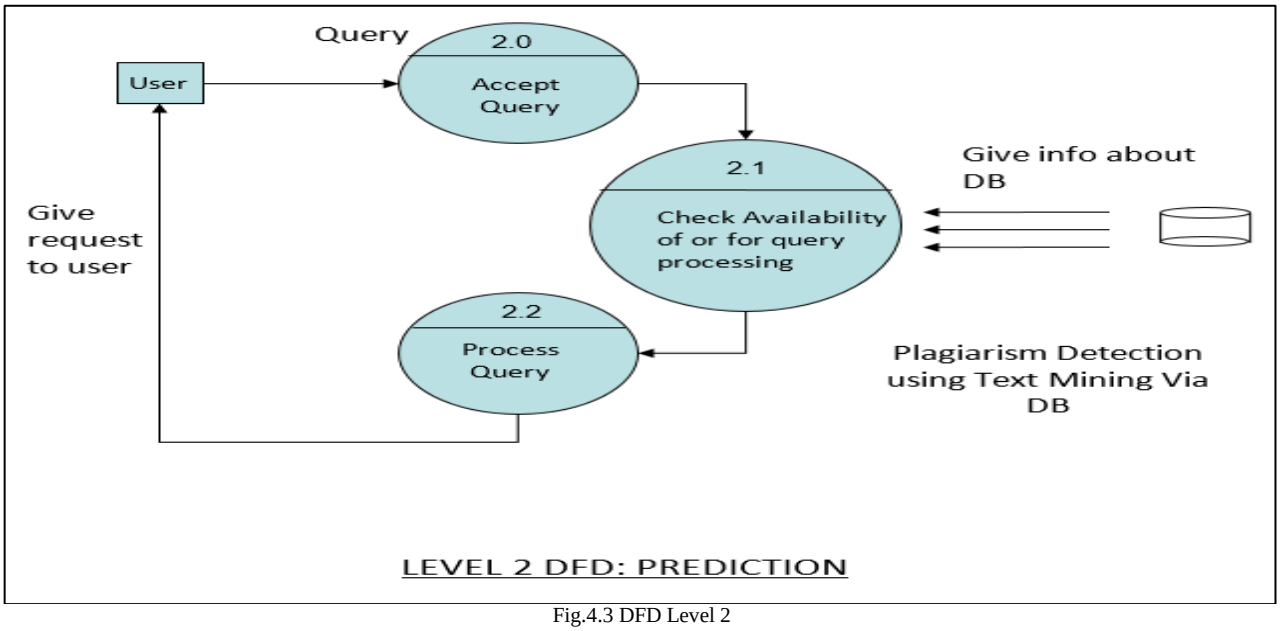

Fig.4.3 DFD Level 2

V. RESULTS

SNAPSHOTS:-

- The website contains 3 login Student, Faculty and Admin.

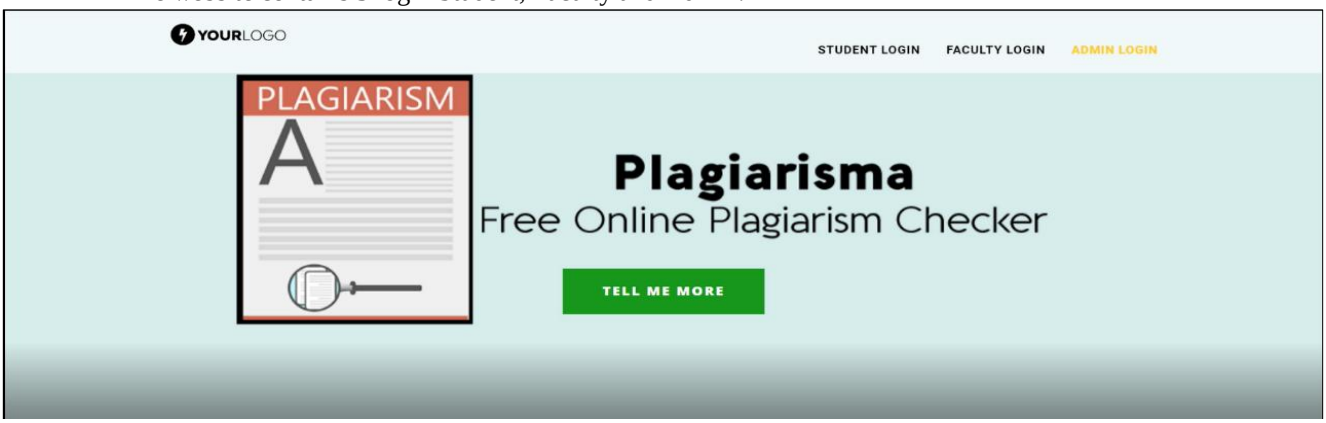

Fig.5.1 Font Page

- Faculty can add and assign topic for the students to be submitted.

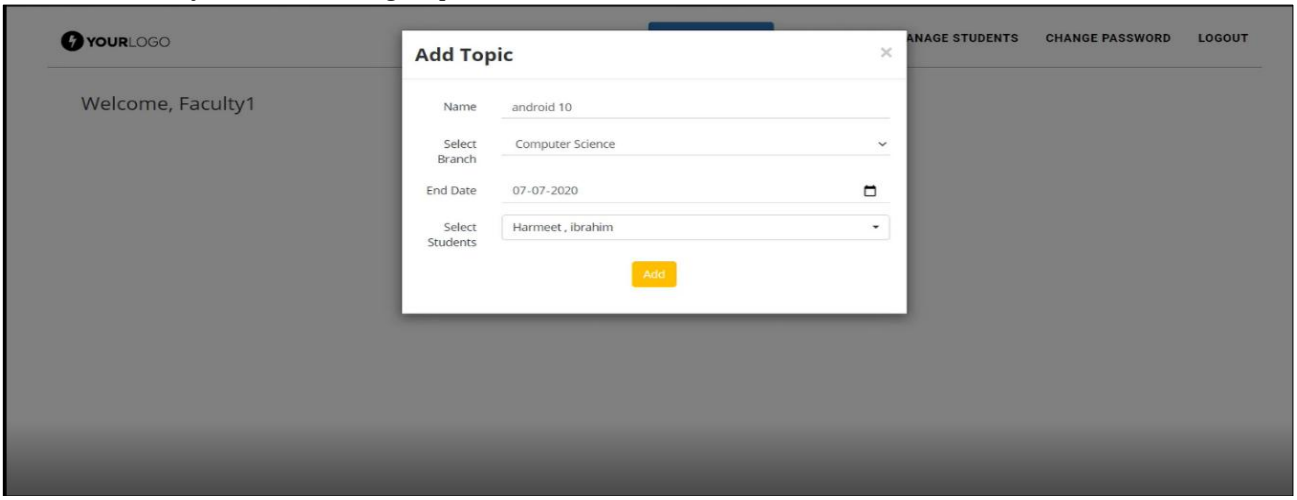

Fig.5.2 Faculty Page

- The topic will be appeared on the student login and will ask to submit the assignment andthen the student can check for internal plagiarism.

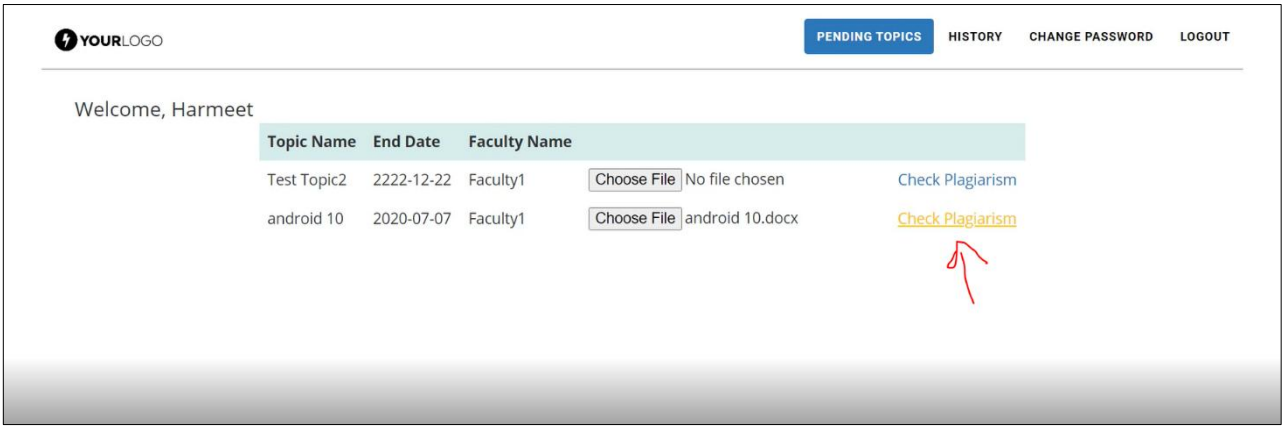

Fig.5.3Student Page

- The internal and external plagiarism is checked by the clicking on it .

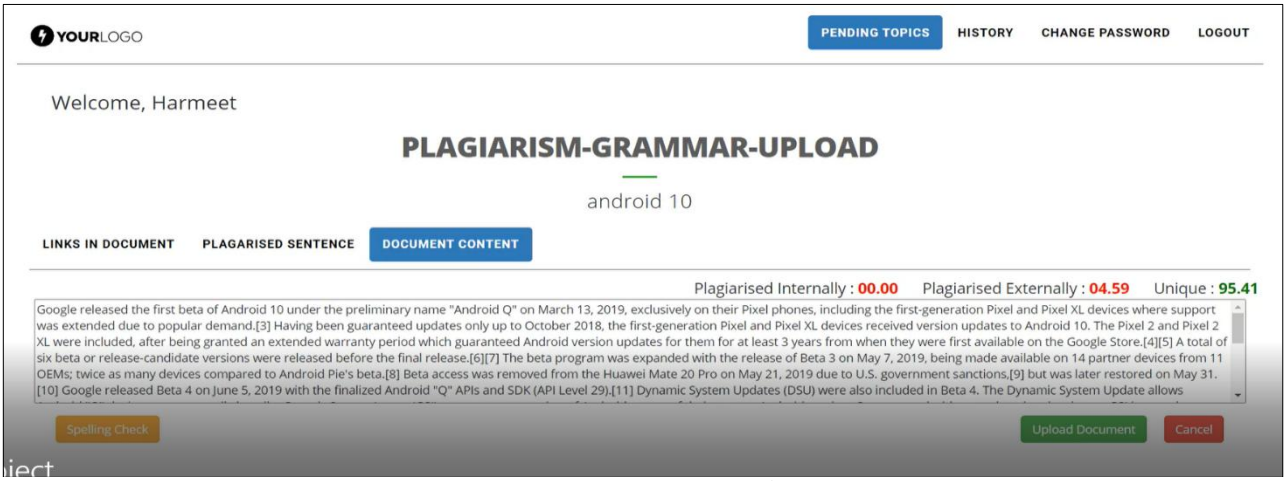

Fig.5.4Final Output Of Student

- The plagiarism is checked by the faculty in the faculty loginas it shows the plagiarized value in the form of percentage of all the students.

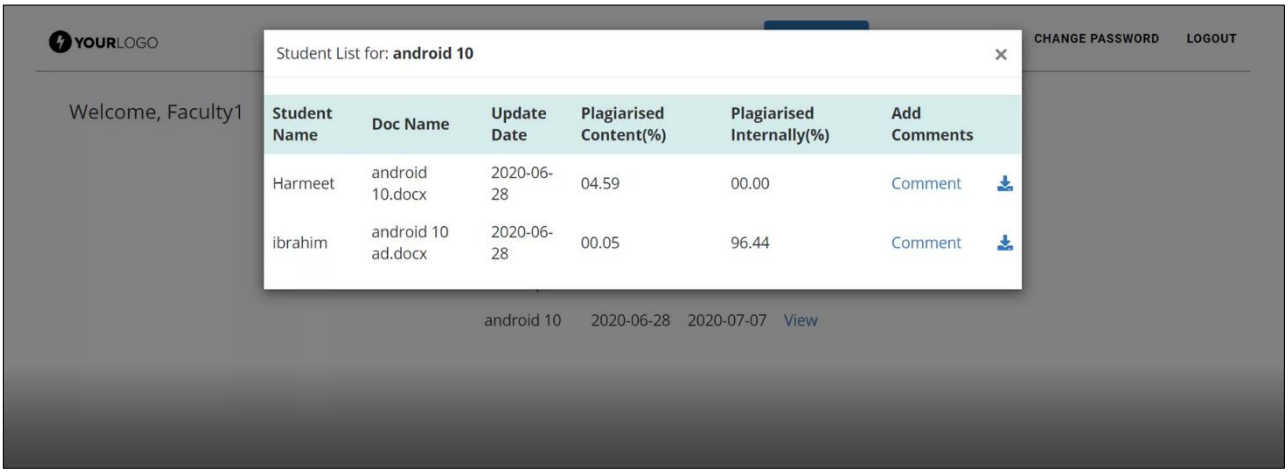

Fig.5.5 Final Output Of Teacher

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

||Volume 8, Issue 6, June 2020||

VI. CONCLUSION AND FUTURE WORK

The software prepared is able to search words that are plagiarized and give the match score according to similarity of original content. This was our project of System Design about "**Online Plagiarism Checker using Text Mining**" developed using Visual Studio as web application based on Asp .Net with C# language. The Development of this system takes a lot of efforts from us. We think this system gave a lot of satisfaction to all of us. Though every task is never said to be perfect in this development field even more improvement may be possible in this application. We learned so many things and gained a lot of knowledge about development field. We hope this will prove fruitful to us.

REFERENCES

[1] Patrick M. Scanlon "Student Online Plagiarism" ,Vol. 51, No. 4 (Fall, 2003), pp. 161-165
[2] Culwin, F. & Lancaster, T. (2001) 'Plagiarism issues for higher education', Vine, 123, pp. 36-41.
[3] Aaron, R. M. (1992). Student academic dishonesty: Are collegiate institutions addressing the issue? NASPA Journal, 29, 107-113.
[4] Nuss, E. M. (1984). Academic integrity: Comparing faculty and student attitudes. Improving College and University Teaching, 32 (3), 140-144.
[5] Wilhoit, S. (1994). Helping students avoid plagiarism. College Teaching, 42(4), 161-164.
[6] Atkins, Thomas, & Nelson, Gene. (2001). Plagiarism and the Internet: Turning the tables. English Journal, 90(4), 101–104.
[7] Grijalva, T., Nowell, C., & Kerkvliet, J. (2006). Academic honesty and online courses. College Student Journal, 40(1), 180-185.
[8] Dant, D. (1986). Plagiarism in high school: A survey. English Journal, 75(2), 81–8
[9] Roig, M. (2001). Plagiarism and paraphrasing criteria of college and university professors. Ethics and Behaviour, 11(3), 307–324.
[10] Malloch, A. E. (1976). A dialogue on plagiarism. College English, 38, 65–74.