"In Pursuit of Technical Excellence"

PROJECT REPORT

On

"Advanced Plagiarism Tool"

For the Degree of

Bachelor of Technology in
Information Technology

By

Prashant Kadam	**BE20F06F026**
Anjali Dhole	**BE21S06F002**
Abhishek Joshi	**BE21S06F003**
Vedant Wagh	**BE21S06F009**

Under the Guidance of

Dr. Shilpa Kabra

Department of Information Technology

GOVERNMENT COLLEGE OF ENGINEERING, AURANGABAD

CHHATRAPATI SAMBHAJINAGAR

(An autonomous Institute of Government of Maharashtra)

(2023-2024)

CERTIFICATE

This is to certify that the seminar entitled **"Advanced Plagiarism Tool"** which is being submitted herewith for the award of the **'Degree of Bachelor of Engineering' in 'Information Technology'** of Government College of Engineering, Aurangabad (Chhatrapati Sambhajinagar) by **Mr. Prashant Kadam, Miss Anjali Dhole, Mr. Abhishek Joshi and Mr. Vedant Wagh.**

Place: Chhatrapati Sambhajinagar

Date:

Dr. Shilpa Kabra
Project Guide

Dr. Anjana Ghule
Head of Department

Dr. Sanjay Dambhare

Principal

Government College of Engineering,

Aurangabad (Chhatrapati Sambhajinagar)

INDEX

List of Figures:

List of Abbreviations:

Abbreviations	**Description**
NLTK	Natural Language Toolkit
CS	Cosine Similarity
DOC	Documents
CAL	Calculate
BOW	Bag Of Words
CV	Count Vectorized
TF	Term Frequency
IDF	Inverse Document Frequency
TF*IDF	TF-IDF Multiplier
LSA	Latent Semantic Analysis
SVD	Singular-Value Decomposition
OTM	One to Many
OTO	One to One
AD	After Deadline
BD	Before Deadline

CHAPTER 01 INTRODUCTION

1.1 Introduction:

All writers and authors use words and ideas borrowed from other sources. Journalists use facts and data they discover in their research. A novelist might use some idea she or he read in another book and then plot it by manipulating it. Poets regularly borrow words, images, and metaphors. Academic writing is not that different. Whether the author is a chemist writing about a new discovery in the lab, a sociologist describing a new theory, or an English literature professor writing about Shakespeare, academic writers usually make heavy use of previous writing on the same topic.

However, one important difference between academic writing and other genres of writing is the importance of indicating the sources where words and ideas were borrowed from. No one expects a poet to footnote a poem to indicate where she or he found the words and metaphors. Identifying the source of words and ideas really holds very much significance. In the tradition of academic writing, originality is parameter. Identifying sources is so vital in the tradition of academic writing that to not identify or specify your sources is considered a 'crime': the crime of plagiarism.

In the academic world, plagiarism by students is usually considered a very serious offense that can result in punishments such as a failing grade on the particular assignment, the entire course, or even being expelled from the institution. For cases of repeated plagiarism, or for cases in which a student commits severe plagiarism (e.g., purchasing an assignment), suspension or expulsion may occur.

However, to impose sanctions and considering the originality as a parameter, plagiarism needs to be detected.

1.2 What is Plagiarism Detection:

According to the definition of Oxford Languages, Plagiarism the practice of taking someone else's work or ideas and passing them off as one's own.

Plagiarism Detection is the process of pointing out instances of plagiarism and/or copyright infringement within a work or document. The widespread use of computers and Internet have made it easier to plagiarize the work of others.

Plagiarism detection has two types:

1. Mono-lingual detection:

Mono lingual emphasizes on comparing and locating the instances only within the sources of the same language.

2. Cross lingual detection:

Cross lingual detection emphasizes on comparing and locating whether the text in one language is plagiarized with the text written in another language.

Plagiarism is detected by using various techniques and algorithms. The detection tool we have developed focuses on using an existing algorithm according to our objectives and then implementing it by developing modules on certain platform. The focus of our tool is Mono lingual detection. The tool we are developing is Offline Plagiarism Tool

1.3. Need Of Advanced Plagiarism Tool

- To avoid any intentional or unintentional Plagiarism.
- To originalize work by getting the similarity score using such tools To validate the document that is it really acceptable by the organizations.
- To avoid the paraphrasing mistakes.

1.4. Advantages of Advanced Plagiarism Tool

- It helps in increasing the paraphrasing skills as well as the skill of organizing the content.
- It displays various graphs so that the user gets to understand the results easily.

1.5. Outline Of Report

1.5.1 Introduction

The chapter 1 contains the project topic along with its brief explanation and detailed concepts used.

1.5.2 Literature Survey

The chapter 2 reviews the related literature in the various techniques algorithm that are use like Cosine Similarity, Latent Semantic analysis its paper that are referred for presenting project. It covers various technical aspects along with papers and journals.

1.5.3 System Design and Development

The chapter 3 explores the detailed system which describes the working of system along with its architecture, modules and various diagrams of System Design.

1.5.4 Implementation Details of Plagiarism Tool

The chapter 4 describes the detailed implementation of the modules that are developed for the system. It defines and describes the algorithms and techniques that are being implemented in the system.

1.5.5 performance Analysis and Result

The chapter 5 gives the results of the system after implementation along with results. Consequently, the performance of system is evaluated.

1.5.6 Conclusion

The chapter 6 concludes with the remarks on presented project. At last, the future scope for the project is presented.

CHAPTER 02 LITERATURE SURVEY

2.1. Related work

In year 2015, Saprativa Bhattacharjee, Anirban Das, Ujjwal Bhattacharya, Swapan K. Parui and Sudipta Roy proposed “Sentiment Analysis using Cosine Similarity Measure”. This study referred a novel idea based on Cosine Similarity measure which was proposed for classifying the sentiment expressed by a user’s comment into a five-point scale of -2 (highly negative) to +2 (highly positive). The performance of the proposed strategy was compared with some of the wellknown machine learning algorithms namely, NaiveBayes, Maximum Entropy and SVM.

The proposed Cosine Similarity based classifier gives 82.09% accuracy for the 2-class problem of identifying positive and negative sentiments. It outperforms all other classifiers by a considerable margin in the 5-class sentiment classification problem with an accuracy of 71.5%. The same strategy is also used for categorizing each user comment into six different Telecom specific categories. The major contributions of this work are: (a) development of a benchmark dataset for sentiment analysis in the Telecom domain, (b) annotation of the dataset with degree of sentiment on a five-point scale, (c) development of a robust lexicon-based algorithm for preprocessing of noisy text and (d) development of an efficient classification scheme which can be used for both sentiment analysis as well as text categorization.[1]

In the paper “Latent Semantic Analysis: An Approach to Understand Semantic of Text” by Ms. Pooja Kherwa1, Dr. Poonam Bansal they studied latent semantic analysis based on single value decomposition. The aim of Latent semantic analysis is to exploit the global structure of documents. The emphasis of latent semantic analysis is to find hidden relationship in document for better understanding the relationship between terms and document in dataset. They have conducted a study using Latent semantic analysis (LSA) to find correlation of terms in a dataset consisting of research papers of various natural language processing applications. LSA shows that single value decomposition collapses multiple terms with same semantic and can identify terms with multiple meaning and represent documents in lower dimensional conceptual space. The study concluded even though, the latent semantic analysis dimension reduction establishes correlation between terms, the latent semantic analysis is causing a degradation in the correlation of a term to itself.[2]

In this paper “Network-Based Bag-of-Words Model for Text Classification” by Dongyang Yan, Keeping Li, Shuang Gu, Liu Yang they proposed a network-based bag-ofwords

model, which collects high-level structural and semantic meaning of the words. Because the structural and semantic information of a network reflects the relationship between nodes, the proposed model can distinguish the relation of words. They applied the proposed model to text classification and compare the performance of the proposed model with different text representation methods on four document datasets. The results showed that the proposed method achieves the best performance with high efficiency. They proposed the AEBoW (Average Embedding Bag of Words) model based on the complex network to represent text. The AEBoW is an improvement on the BoW model, taking the correlation of words reflected in the text network into consideration.[3]

In the paper "Integrating Collocation as TF-IDF Enhancement to Improve Classification Accuracy" by Gleen A. Dalaorao , Ariel M.Sison , Ruji P. Medina their study aimed to enhance TF-IDF by integrating collocation as a term feature. The collocated terms are extracted based on the determination of part-of-speech (POS) that forms specific patterns such as adjective + noun, noun + noun, noun + verb, etc. The result of this experiment shows that integrating collocation as part of the enhancement of the TF-IDF process outperforms the traditional TFIDF by an increase of up to 10 percent. Their successfully demonstrated that considering collocation in the TF-IDF enhancement has proven to improve the classification accuracy.[4]

In the paper "A Natural Language Processing Framework for Assessing Hospital Readmissions for Patients with COPD by Ankur Agarwal, Christopher Baechle, Ravi Behara, Xingquan Zhu" they proposed a framework which uses Natural Language Processing to analyze clinical notes and predict readmission. Many algorithms within the field of data mining and machine learning exist, so a framework for component selection is created to select the best components. Naïve Bayes using Chi-Squared feature selection offers an AUC of 0.690 while maintaining fast computational times. The readmission analysis system represents a natural language approach to patient readmission prediction. Components were evaluated and it was found that using NB classifier with CS, selecting around 15% of the full feature set to be most effective.[5]

In [6], the author proposed a user interface that would allow users to make comparisons and take documents as inputs. It also offers one of the most effective processing of documents flowcharts that can be found.

The author in [7] proposes a few natural language processing techniques, such as corpusbased, knowledge-based, and string-based approaches, for the semantic analysis of the document.

Semantic meaning is the primary basis for classification of texts.

A text summarizing technique to extract only relevant material is defined in this research [8]. The document under comparison contains enormous amounts of data, however first we must preprocess the data. It offers a few text extraction techniques to ensure that only pertinent and significant content is compared.

An intelligent method for identifying semantic plagiarism in scientific publications is suggested in this study [9]. A corpus containing the text of original scientific articles has been developed in order to compare suspicious documents with it and identify instances of plagiarism. Using the Mini-Batch K-Means clustering algorithm, the documents are grouped into many groups and then placed into a designated category.

The technique to cluster words from sentences and group comparable ones is covered in this study [10]. The three most popular approaches for identifying phrase similarity are vectorbased, word-to-word, and structure-based.

In order to determine similarity, this paper [11] identified certain drawbacks with text clustering and suggested an alternative approach. These restrictions will be addressed by utilizing XLNet in conjunction with a DKM-clustered Bi-LSTM model. The generalized autoregressive model, which creates the highlighted vectors from preprocessed data, is the main contribution of the presented study.

Similarly, word embedding is used in [12] to create the cosine similarity algorithm for word-to-word mapping detection.

Paper [13] provides a definition. Latent semantic analysis (LSA) is a technique that uses certain mathematical calculations to analyze text and examine the relationships between terms and documents within a corpus. Singular value decomposition is used to break down the corpus of related words into manageable numerical representation. LSA demonstrates how a single value decomposition may recognize terms with numerous meanings, collapse several terms with the same semantic, and represent texts in a conceptual space which is in lower dimensional.

This study [14] examines Along with the traditional text processing procedure in semantic analysis, the most popular models and techniques for semantic text processing are taken into account.

2.2. Existing System

There are many offline plagiarism tools and checkers available on the internet; as name suggests plagiarism checker every tool accomplish this task by various method and ways.

Earlier when in the academic culture, plagiarism was considered as a serious offence which would result in expulsion or failure. This gave a boom to checking plagiarism and consequently to develop such checkers and tools

There are systems who have some sort of classrooms or groups created wherein the user has to upload their work before deadline and as soon as deadline passes, they get to see their results of similarity score.

Then on the top this there are instant checkers available u provide an option of uploading file as well as copying the text into text area and then within few seconds results are obtained. Not only this there are also some desktop applications which can compare the documents within the system and provide accurate results.

This all study of various tools brings to the conclusion that there should be systems which will help in carrying out comparison between the specific domain concerned documentations and files in order to main the originality of work within the organization or any institute.

2.3. Objectives

Our Offline Plagiarism Tool mainly aims to:

- Implementing the existing algorithm according to our systems precise needs.
- To get the accurate results.
- To generate the similarity, score not only considering the words as well-meaning and paraphrase.
- To provide a very user-friendly interface that a non-technical person can handle it too.

CHAPTER 03 SYSTEM DESIGN AND DEVELOPMENT

3.1 Requirement Analysis

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. It involves all the tasks that are conducted to identify the needs of different stakeholders. Therefore, requirements analysis means to analyze, document, validate and manage software or system requirements.

The requirements have been elaborated in the following sections.

3.1.1. Normal Requirement

These are the requirements clearly stated by the customer hence these requirements must be present for the customer satisfaction.

NR1: Should give the proper similarity percentage.

NR2: Should accept files with various extensions

3.1.2. Expected Requirement

These requirements are implicit type of requirements, these requirements are not clearly stated by customer but implicitly come during system design.

E1: Should compare file with single selected file as well as all the files in the database. E2: Interface should be user friendly.

3.1.3. Excited Requirement

These requirements are neither stated by customer nor expected. But to achieve total customer satisfaction the developer may include some requirements which enhance the functionality of project.

EXR1: The results should be easily readable and understandable. EXR2: Project should be platform independent.

3.2. Validation of Requirements

Validation of Requirements is the process of checking that requirements defined for development, define the system that the customer really wants.

To check issues related to requirements, we perform requirements validation.

3.2.1. Validation of Normal Requirement

VN1: The System provides proper similarity percentage.

VN2: Accepts Files with extensions (.pdf, .txt, .doc, .docx)

3.2.2. Validation of Expected Requirement

VE1: System Provides option to the user to select a file to be compared with. **VE2:** Any non-technical person can handle the interface

3.2.3. Validation of Excited Requirement

VX1: System provides results in graphical format (BAR GRAPH, DOUGHNUT-PIE CHART)

VX2: System runs on any Platform

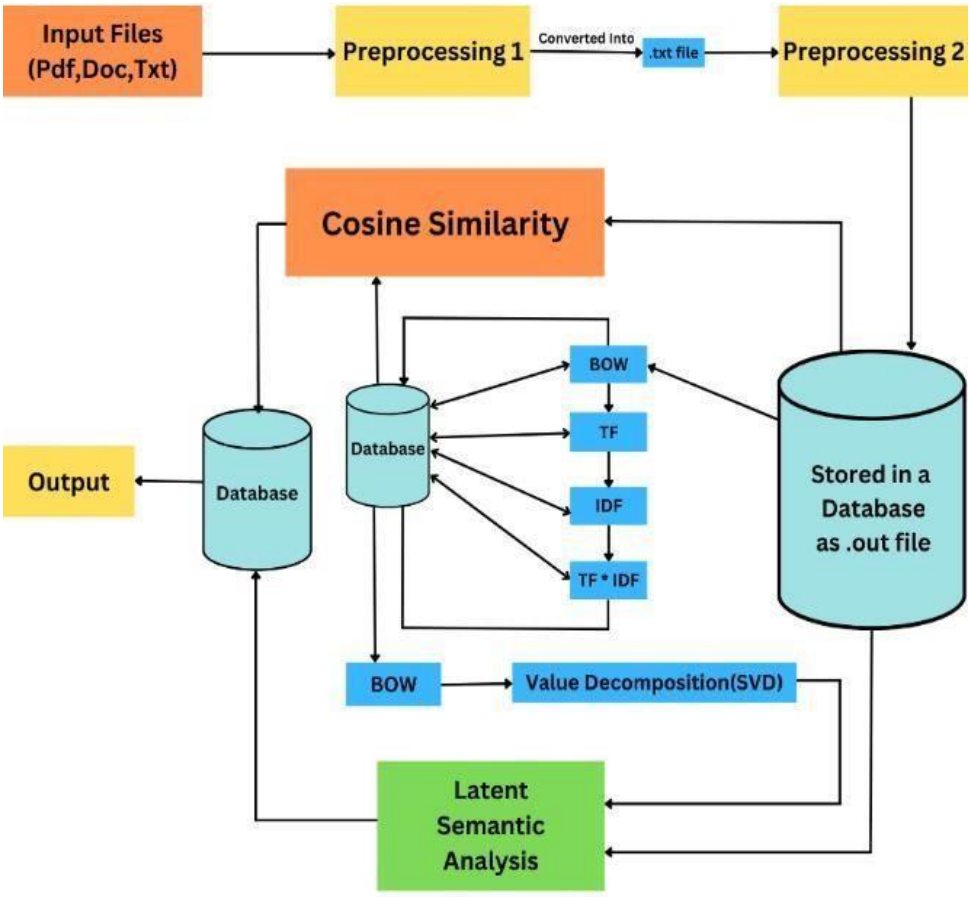
3.3. Hardware Requirement

Content	**Description**
Main Processor	Pentium IV Intel i3, i5, i7.
RAM	1 GB Minimum 2 GB Recommended
Hard Disk Drive (HDD)	40 GB Minimum 1 TB Recommended
Ports	1 Serial Port

3.4. Software Requirement

Content	Description
Platform	Platform Independent
Language	PYTHON, HTML, CSS, JAVASCRIPT
Database	Sqlite3
Server	Localhost
Framework	Bootstrap, Flask

3.5. Architecture



The architecture of system shows the flow of execution in proper manner.

Firstly, the files are accepted then they are pre-processed and converted to .txt format later on the file is passed to next block in order to extract only the concerned data to be used ahead. Secondly this data is stored into the database and then all the modules get triggered wherein the algorithms are implemented. lastly all the results are calculated and stored which is then retrieved on a graphical user interface.

3.6. System Design

System design is the process of defining architecture, modules, interface and data for a system to satisfy the specified requirements. All the details are elaborated in the following section.

3.6.1 Breakdown Structure

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, and plan, improve and communicate often complex processes in clear, easy-to-understand diagrams.

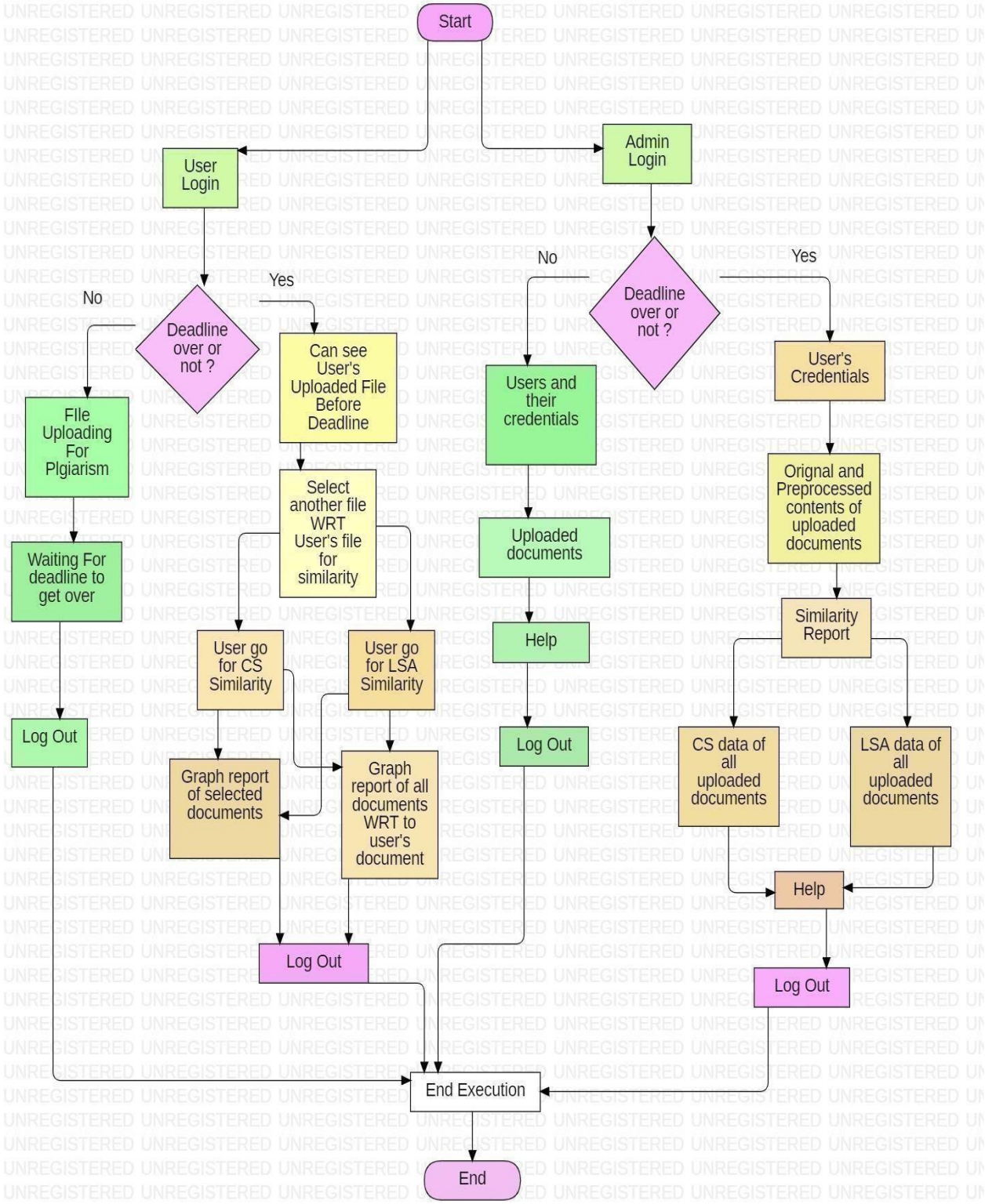
1. Procedure User Login Phase

The aim of this phase is taking the users credentials and allow him to logged in his account if credentials are valid. This phase is divided into two sections Before Deadline and After Deadline. User can upload the document for checking similarity purpose before deadline.

Once deadline gets over user will not able to upload any documents. But After Deadline when user will log in his account, he will get plagiarism report of his previously uploaded document and can also see the graph percentage with respective one. And finally, can logged out.

2. Procedure Admin Login phase

Admin has some administrative rights. Admin also has to face to sub phases i.e. Before Deadline and After Deadline. If Admin logged in Before Deadline, then he can see registered users, their ID and Passwords and files uploaded by them as well as some project credentials. As the similarity report will get ready After Deadline Admin can see the similarity report of all the documents of all registered users.

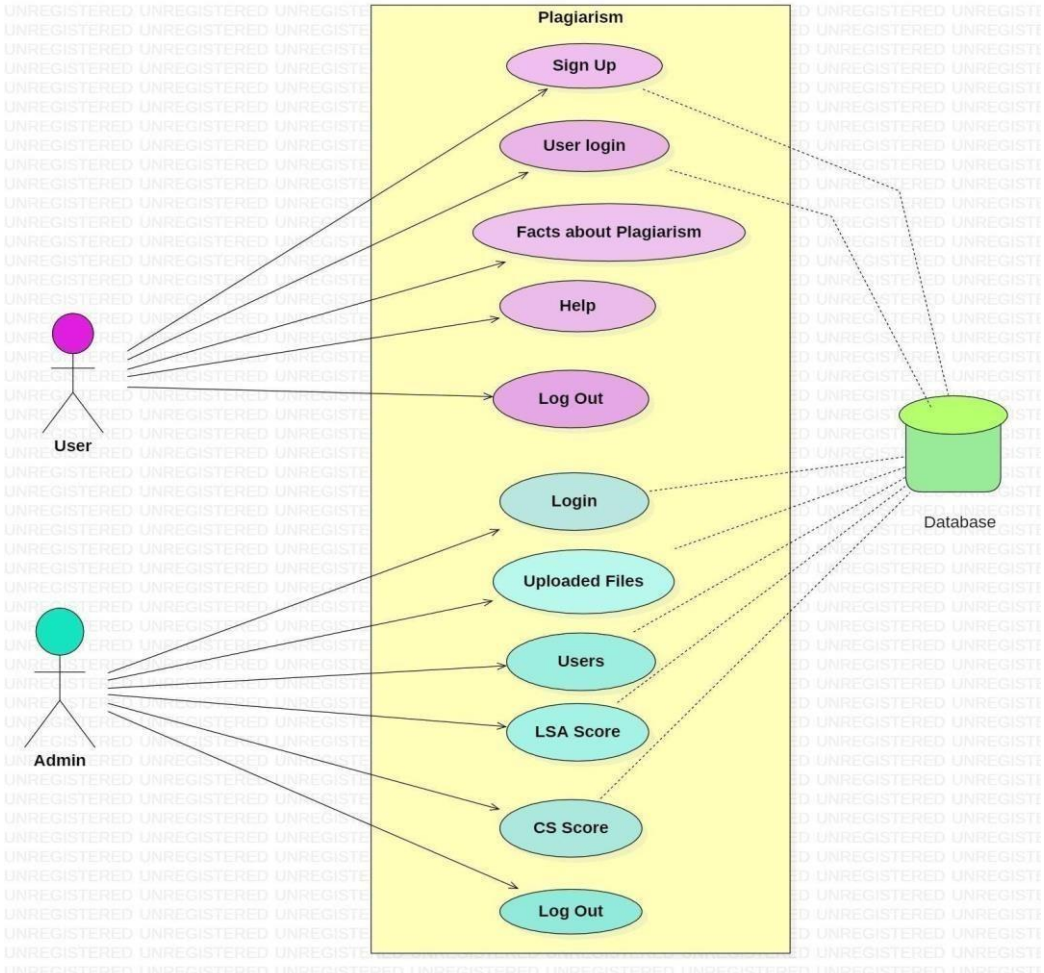

Start
User Login
Admin Login
No
Yes
Deadline over or not ?
Can see User's Uploaded File Before Deadline
FIle Uploading For PIgiarism
Waiting For deadline to get over
Log Out
Select another file WRT User's file for similarity
User go for CS Similarity
User go for LSA Similarity
Graph report of selected documents
Graph report of all documents WRT to user's document
Log Out
No
Yes
Deadline over or not ?
Users and their credentials
Uploaded documents
Help
Log Out
User's Credentials
Orignal and Preprocessed contents of uploaded documents
Similarity Report
CS data of all uploaded documents
LSA data of all uploaded documents
Help
Log Out
End Execution
End

3.6.2 Use Case Diagram

A UML use case diagram is the primary form of system/software requirements for a nytuttew software program underdeveloped. Use cases specify the expected behaviour (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modelling is that it helps us design a system from the end user's perspective.

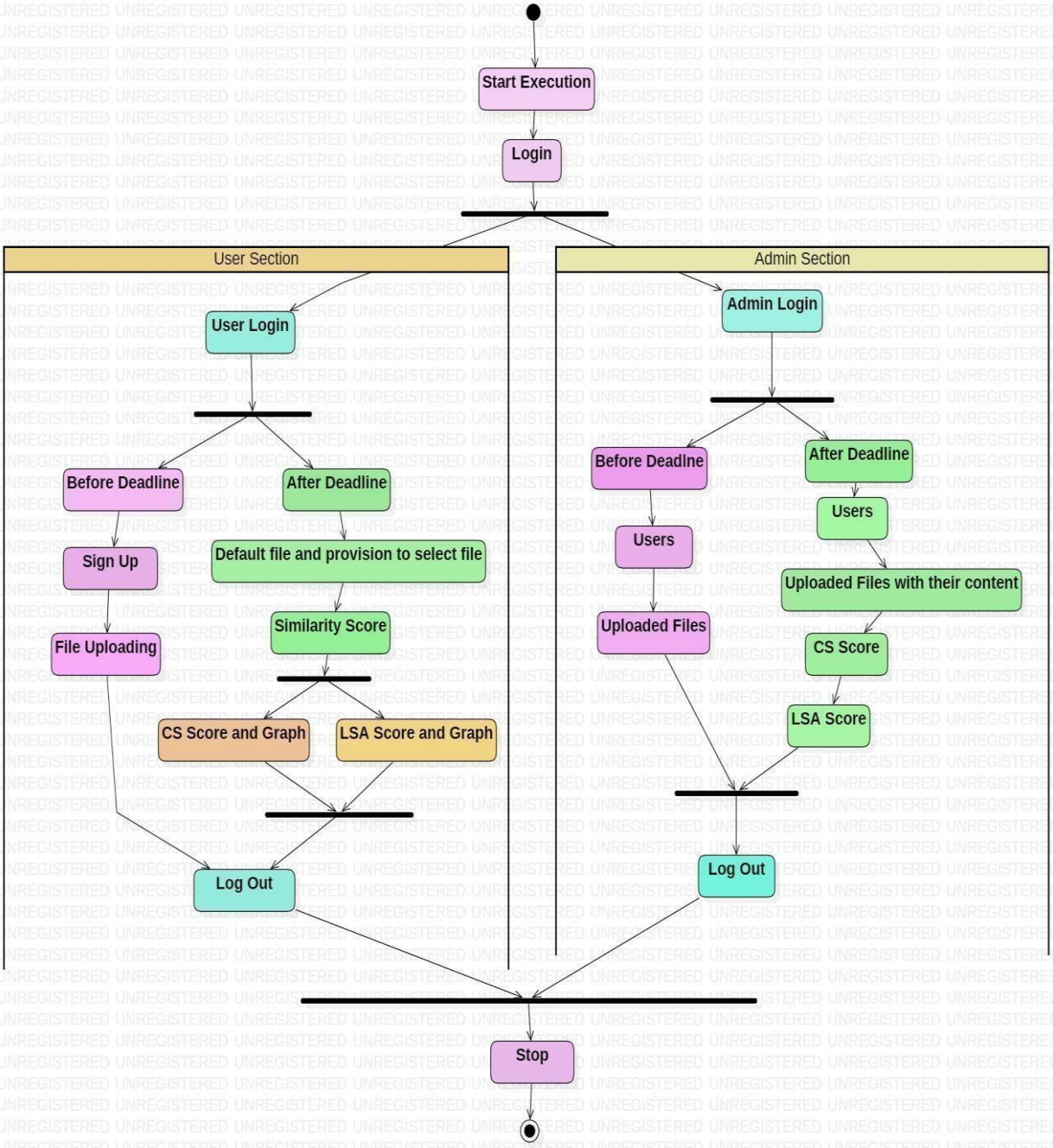
1. **Procedure User Registration Phase**

This is the starting phase of our project. The aim of this phase is to register the newly coming users to our tool. For the document uploading and plagiarism checking purpose user has to login with their valid credentials before deadline. So, for the login purpose user has to register first. So that he will get his id and password which is helpful for login purpose.



3.6.3 Activity Diagram

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination.



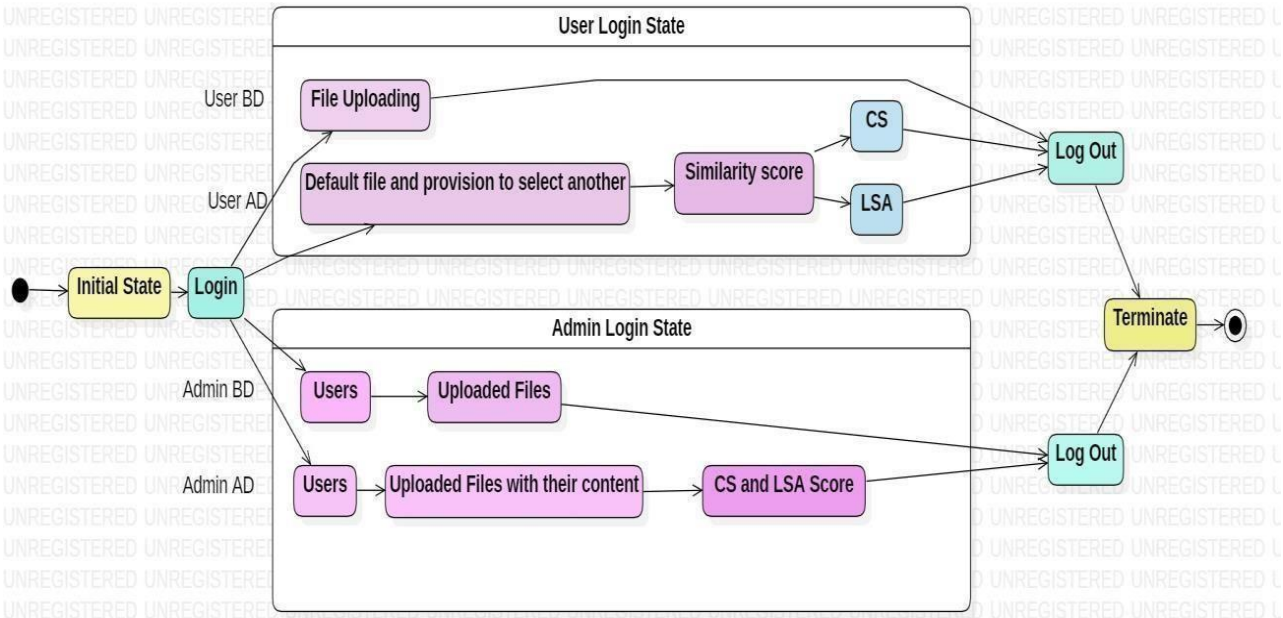
CHAPTER 04 IMPLEMENTATION DETAILS

4.1. Introduction

This chapter elaborates the implementation of the system. The break down structure of the system is already described in previous chapter. All the details of modules along with its functionality is explained in the following section:

4.2. State Diagram

State-transition diagrams describe all of the states that an object can have, the events under which an object changes state (transitions), the conditions that must be fulfilled before the transition will occur (guards), and the activities undertaken during the life of an object (actions). State transition diagrams are very useful for describing the behaviour of individual objects over the full set of use cases that affect those objects.



4.3. Preprocessing

Text Preprocessing is a technique in the area of Natural Language Processing (NLP). This technique that involves transforming raw data into an understandable format. For comparison of documents, we need data in the numeric form. We basically used two algorithms (Cosine Similarity, Latent Semantic Analysis) to compare documents. But before comparison, we first need to clean the text data. Hence, we have used preprocessing.

Python Library used for Preprocessing:

Natural Language Toolkit library (NLTK):

- It is a strong and robust library.
- It is a powerful Python library that consists of various natural languages algorithms.
- It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning.
- It has great pre trained models and corpus of data which makes text processing and analysis pretty quick and easy.

4.3.1 Preprocessing techniques:

4.3.1.1. Tokenization:

Tokenization is about splitting strings of text into smaller pieces, or "tokens". Paragraphs can be tokenized into sentences and sentences can be tokenized into words.

Example: "Never give up." Output: {'Never','give','up'}

4.3.1.2. Stop words and punctuation marks removal:

Stop words are very common words. These words as well as punctuation marks do not really signify any importance as they do not help in comparing two documents. Hence, we can remove stop words and punctuation marks to save computing time and efforts in processing large volumes of text.

4.3.1.3. Lemmatization:

Lemmatization is the process of converting a word to its base form. It reduces the inflected words properly ensuring that the root word belongs to the language. Example: Caring-> Care

4.3.1.4. Lower casing:

Lowercasing means Converting a word to lower case (NLP -> nlp).

Words like Book and book mean the same but when not converted to the lower case those two are represented as two different words.

4.4. Cosine Similarity

4.4.1. Bow:

The bag-of-words (BOW) module is a method used in NLP and Information Retrieval.

BOW representation includes two things:

a. a vocabulary of known words,
b. a measure of the presence of known words.

Bow module is one of the important modules as it is used for CS as well as LSA

When new document will be uploaded by user then preprocessing will be applied. All text will split into words then bow module will be executed. At first, all words will be stored into one globally declared list and then count of each word in document will be calculated. At last, all words and their respective count will be stored into database. These values are then used by TF, IDF and LSA module to perform respective calculations.

4.4.2. TF:

Term Frequency (TF) measures number of times a word occurs in a document.

Formula for calculating TF:

TF (word) = (Count of each word in a document) / (Total number of words in the document).

Example: If a document containing 100 words where word software appears 6 times.

Output: term frequency(software)= 6/100= 0.06

When new document will be uploaded by user then preprocessing and bow module will be executed then TF module will be executed. After first, all words and their count will retrieve from BOW table and total number of words will be calculated then formula of TF gets evaluated. At last, all words and their respective TF value will be stored into database. These values are then used by TF-IDF multiplier module.

4.4.3. IDF:

Inverse document frequency (IDF) measures how significant any word is in the collection of documents.

Formula for calculating IDF:

log (total no of documents/ no of documents with word)

Example: If there are 100 documents where word “DEFEND” appears in 10 documents

Output: IDF(DEFEND)= log (100/10) =1

When new document will be uploaded by user then all the prior modules get executed and as IDF is dependent on the number of documents for each value so the values for each and word is to be updated. Whenever the IDF module gets triggered, the previous table gets deleted and the logic of retrieving the number of documents from BOW as well as occurrence of word gets executed and all the values are substituted to the formula and updated respectively considering the id. These values are then used by TF-IDF multiplier module.

4.4.4. TF-IDF Multiplier:

TF-IDF is text based statistical weighting techniques used for the purpose of information retrieval. It is a method to measure the importance of a term with respect to a document or a collection of documents.

Formula for calculating TF-IDF Multiplier:

TF-IDF Multiplier(word)= TF (word) * IDF (word)

TF-IDF Multiplier is dependent on TF and IDF module. After execution of these 2 modules then TF-IDF Multiplier will be executed. At first, TF values of all words will be retrieved from TF table and IDF values of all words will be retrieved from IDF table. Then formula gets evaluated. At last, all words and their respective TF-IDF value will be stored into database. These values are then used by CS module.

4.4.5. CS:

Cosine similarity is a metric used to determine how similar the documents are irrespective of their size. Cosine similarity measures the similarity between two vectors of an inner product space. It is a similarity rate the calculation obtained from the cosine angle multiplication of two vectors being compared, because the cosine 0 degree is 1 and less than 1 to the value of another angle, then the value of the similarity of the two vectors are said to be similar when the value of the cosine similarity is 1

CS module is dependent on TF-IDF Multiplier. To compare two documents, TF-IDF values of all words of each document will be used as elements of vectors and then formula gets evaluated.

Formula for calculation of Cosine similarity:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

In this formula, a and b are two vectors.

In numerator, 1st element of vector A will be multiplied with 1st element of vector B then 2nd element of vector A will be multiplied with 2nd element of vector B and this goes on till the last element and then their multiplication results will be added.

In denominator, now in the Denominator all the elements of vector A and vector B will be squared and added with the respective vector elements and then their square root will be calculated of the result obtained from 2 vectors and then the results are multiplied.

At last value of numerator will be divided by value of denominator and thus cosine similarity of documents is obtained. Then these values will be stored in Database in diagonal matrix format.

4.5. Latent Semantic Analysis (LSA)

Latent semantic analysis (LSA) is a technique in the area of Natural processing (NLP). The main goal of LSA is to create vector-based presentation for texts to make semantic content. By vector representation, LSA computes the similarity between texts between two or more documents. In the past LSA was named as Latent semantic indexing (LSI) but improved for information retrieval taking. So finding few documents that close to the query that given from many documents.

LSA should have many aspects to give approach such as key words matching, Wight key words matching and vector representation depends on occurrences of words in documents. Also, Latent semantic analysis (LSA) uses singular value decomposition (SVD) to rearrange the data. SVD is a method that uses matrix to reconfigure and calculates all the diminutions of vector space. In addition, the damnations In vector space will be computed and organized from most is the least important in LSA the most significant assumption will be used to fine the meaning of the text otherwise least important will be ignored in the assumption.

By searching about words that have a high rate of similarity will be occurred if that wards have similar vector. To describe the most essential steps in LSA first collecting a huge set of relevant text then divide it by documents Second make co-occurrence matrix for terms and documents also giving the cell name such as documents x, terms y and m for dimensional value for terms and n dimensional vector for documents. Third each cell will be whetted and calculated finials SVD will play big roil to compute all the diminutions and make three matrices.

LSA module is dependent on BOW module. All words and their respective count will be retrieved from BOW table to find their semantic meaning and compare two or more documents.

Python Library used for calculation of LSA:

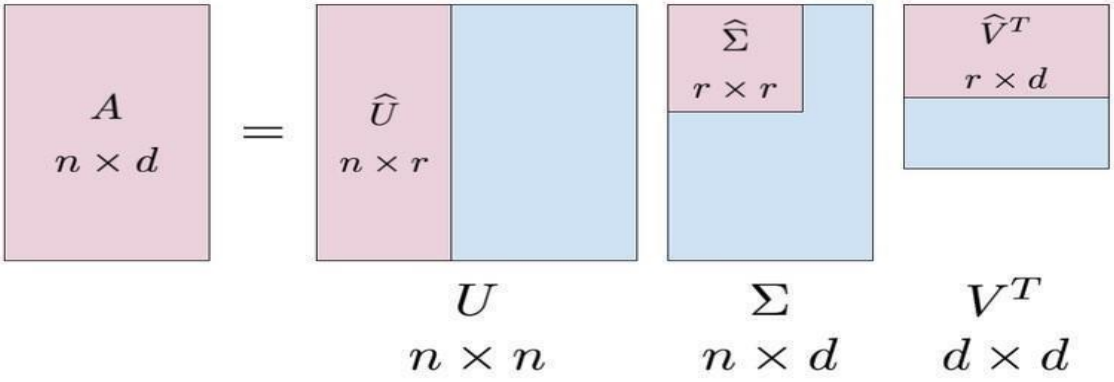
Scikit-learn (Sklearn):

- Scikit-learn is the most useful and robust library for machine learning in Python.
- This library is written in Python.
- It is built upon NumPy, SciPy and Matplotlib.
- It consists of main two main methods which is used to calculate LSA. Methods are:

1. **TruncatedSVD() :**

This transformer performs linear dimensionality reduction by means of truncated singular value decomposition (SVD). In particular, truncated SVD works on term bow matrices as retrieved from BOW table. TruncatedSVD () method reduces computation complexity and gives more relevant and useful results. TruncatedSVD performs SVD function on BOW and gives us vector after dimensionality reduction.

It reduces the dimension like this:



2. **Normalizer(Parmeter1.Parameter2) :**

This method accepts two parameters. parameter1 is all words from BOW and parameter2 is Decompose matrix of BOW. This method returns an array with inbuilt topics and their values. Array contains topic wise word collection and its semantic value.

At last, elements of array will be stored in database. All values in database will be stored in diagonal matrix format.

4.6. Graphical Representation of Similarity Analysis

A chart is a graphical representation for data visualization, in which "the data is represented by symbols, such as bars in a bar chart, lines in a line chart, or slices in a pie chart". A chart can represent tabular numeric data, functions or some kinds of quality structure and provides different info.

Graphs used for representing Similarity Percentage of Documents are:

4.6.1. Bar Graph:

A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally.

Purpose: To represent Similarity percentage of User's Document with respected to all documents.

4.6.2. Doughnut Chart:

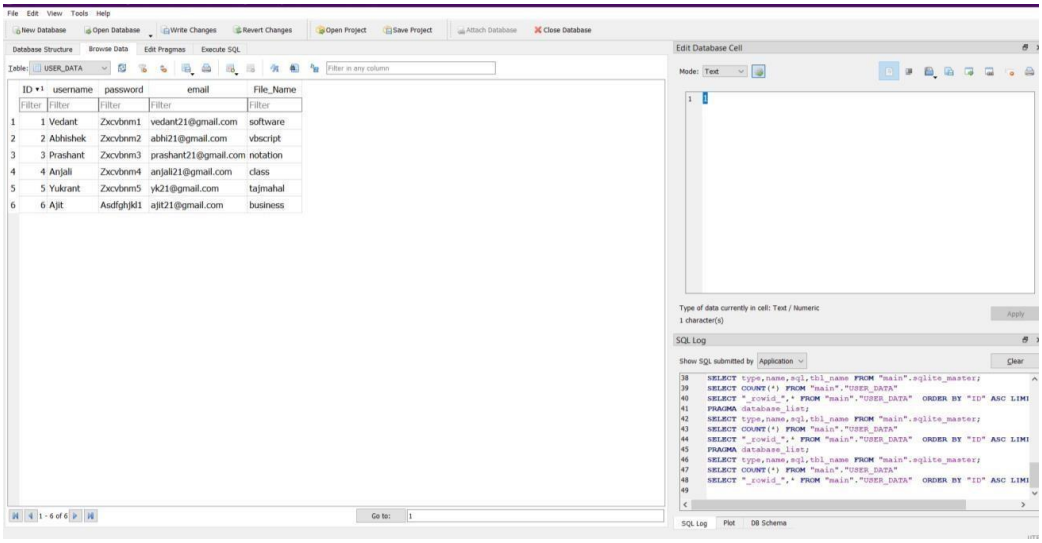
Doughnut charts are the modified version of Pie Charts with the area of centre cut out. A doughnut is more concerned about the use of area of arcs to represent the information in the most effective manner instead of Pie chart which is more focused on comparing the proportion area between the slices.

Doughnut charts are more efficient in terms of space because the blank space inside the doughnut charts can be used to display some additional information about the doughnut chart.

Purpose: To represent Similarity Percentage of User's Document with respected to selected Document.

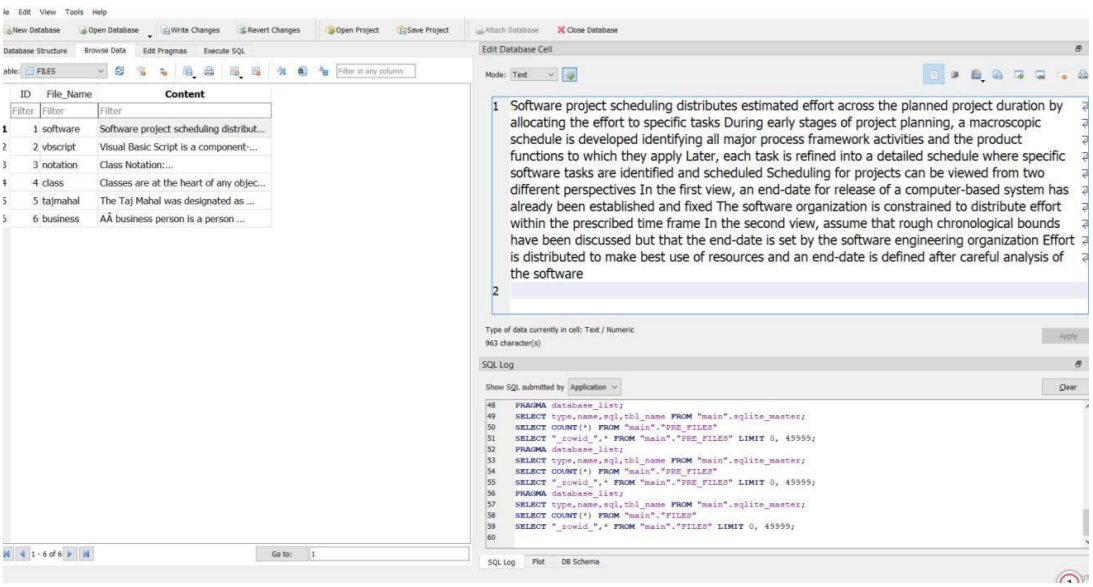
4.7. Screenshots of Databases:

4.7.1. User_Details Database:

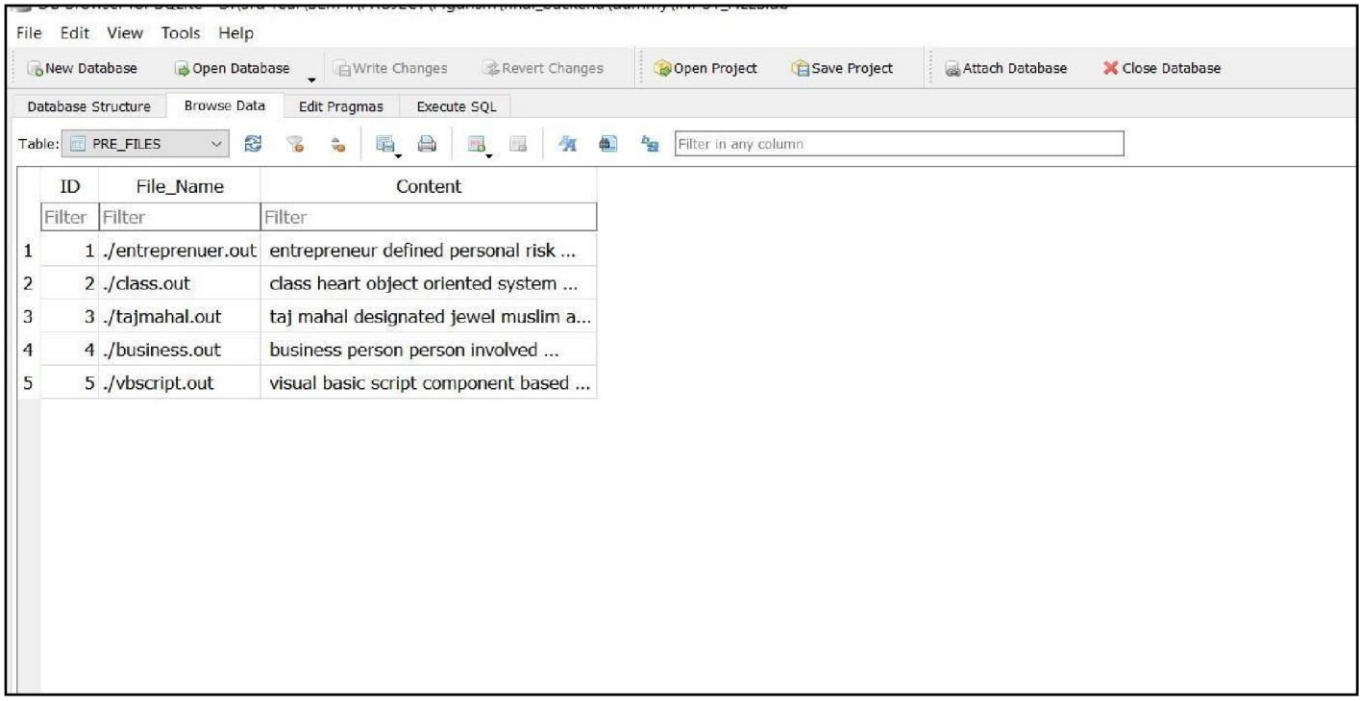


4.7.2 Input_Files Database

4.7.2.1 Files



4.7.2.2 Pre_files table:

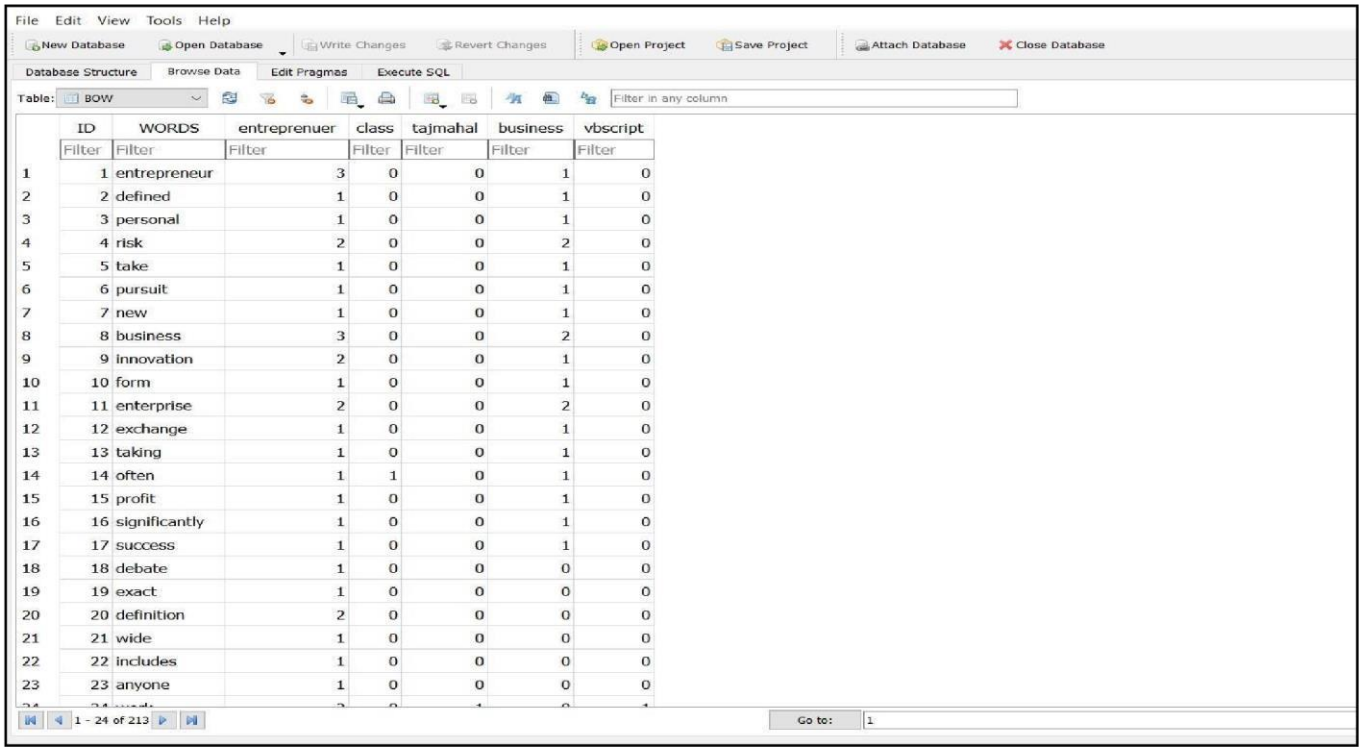


	ID	File_Name	Content
	Filter	Filter	Filter
1	1	./entreprenuer.out	entrepreneur defined personal risk ...
2	2	./class.out	class heart object oriented system ...
3	3	./tajmahal.out	taj mahal designated jewel muslim a...
4	4	./business.out	business person person involved ...
5	5	./vbscript.out	visual basic script component based ...

4.7.3 . Document_Similarity Database :

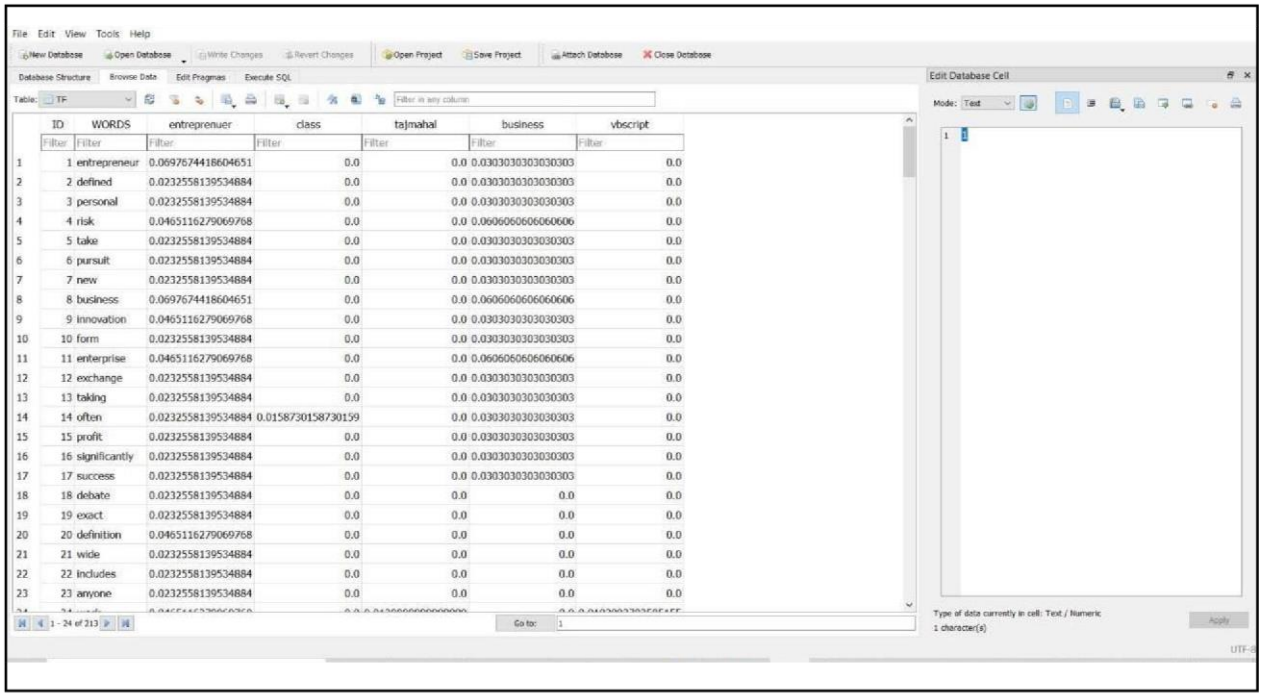
40

4.7.3.1. Bow table:



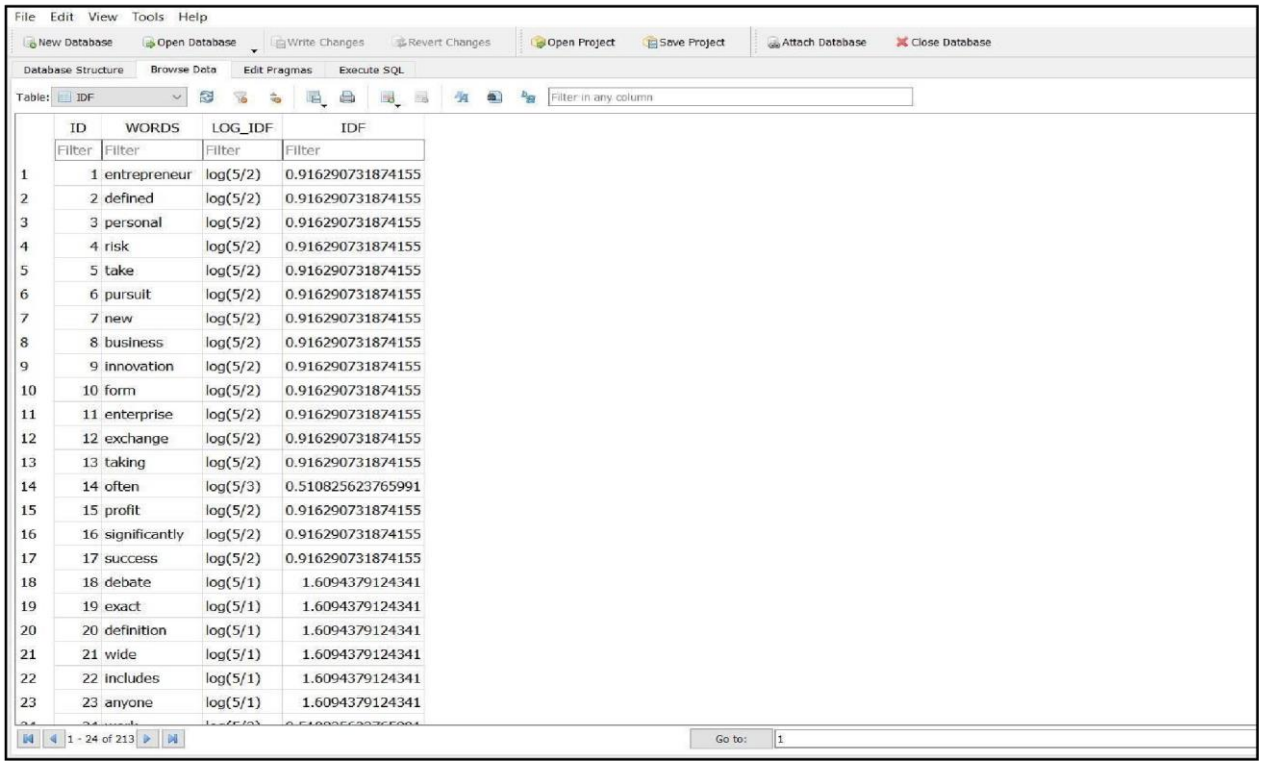
	ID	WORDS	entreprenuer	class	tajmahal	business	vbscript
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	entrepreneur	3	0	0	1	0
2	2	defined	1	0	0	1	0
3	3	personal	1	0	0	1	0
4	4	risk	2	0	0	2	0
5	5	take	1	0	0	1	0
6	6	pursuit	1	0	0	1	0
7	7	new	1	0	0	1	0
8	8	business	3	0	0	2	0
9	9	innovation	2	0	0	1	0
10	10	form	1	0	0	1	0
11	11	enterprise	2	0	0	2	0
12	12	exchange	1	0	0	1	0
13	13	taking	1	0	0	1	0
14	14	often	1	1	0	1	0
15	15	profit	1	0	0	1	0
16	16	significantly	1	0	0	1	0
17	17	success	1	0	0	1	0
18	18	debate	1	0	0	0	0
19	19	exact	1	0	0	0	0
20	20	definition	2	0	0	0	0
21	21	wide	1	0	0	0	0
22	22	includes	1	0	0	0	0
23	23	anyone	1	0	0	0	0

4.7.3.2 .TF table:



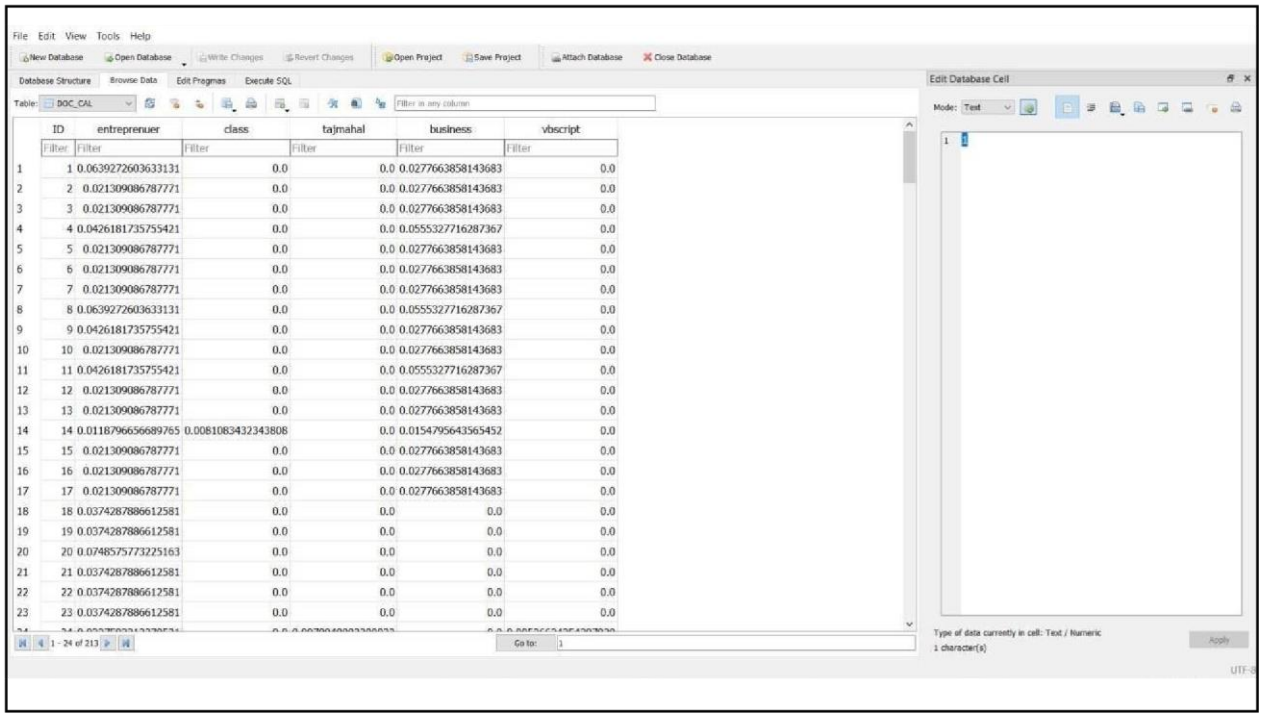
	ID	WORDS	entreprenuer	class	tajmahal	business	vbscript
1	1	entrepreneur	0.0697674418604651	0.0	0.0	0.0303030303030303	0.0
2	2	defined	0.0232558139534884	0.0	0.0	0.0303030303030303	0.0
3	3	personal	0.0232558139534884	0.0	0.0	0.0303030303030303	0.0
4	4	risk	0.0465116279069768	0.0	0.0	0.0606060606060606	0.0
5	5	take	0.0232558139534884	0.0	0.0	0.0303030303030303	0.0
6	6	pursuit	0.0232558139534884	0.0	0.0	0.0303030303030303	0.0
7	7	new	0.0232558139534884	0.0	0.0	0.0303030303030303	0.0
8	8	business	0.0697674418604651	0.0	0.0	0.0606060606060606	0.0
9	9	innovation	0.0465116279069768	0.0	0.0	0.0303030303030303	0.0
10	10	form	0.0232558139534884	0.0	0.0	0.0303030303030303	0.0
11	11	enterprise	0.0465116279069768	0.0	0.0	0.0606060606060606	0.0
12	12	exchange	0.0232558139534884	0.0	0.0	0.0303030303030303	0.0
13	13	taking	0.0232558139534884	0.0	0.0	0.0303030303030303	0.0
14	14	often	0.0232558139534884	0.0158730158730159	0.0	0.0303030303030303	0.0
15	15	profit	0.0232558139534884	0.0	0.0	0.0303030303030303	0.0
16	16	significantly	0.0232558139534884	0.0	0.0	0.0303030303030303	0.0
17	17	success	0.0232558139534884	0.0	0.0	0.0303030303030303	0.0
18	18	debate	0.0232558139534884	0.0	0.0	0.0	0.0
19	19	exact	0.0232558139534884	0.0	0.0	0.0	0.0
20	20	definition	0.0465116279069768	0.0	0.0	0.0	0.0
21	21	wide	0.0232558139534884	0.0	0.0	0.0	0.0
22	22	includes	0.0232558139534884	0.0	0.0	0.0	0.0
23	23	anyone	0.0232558139534884	0.0	0.0	0.0	0.0

4.7.3.3. IDF table:



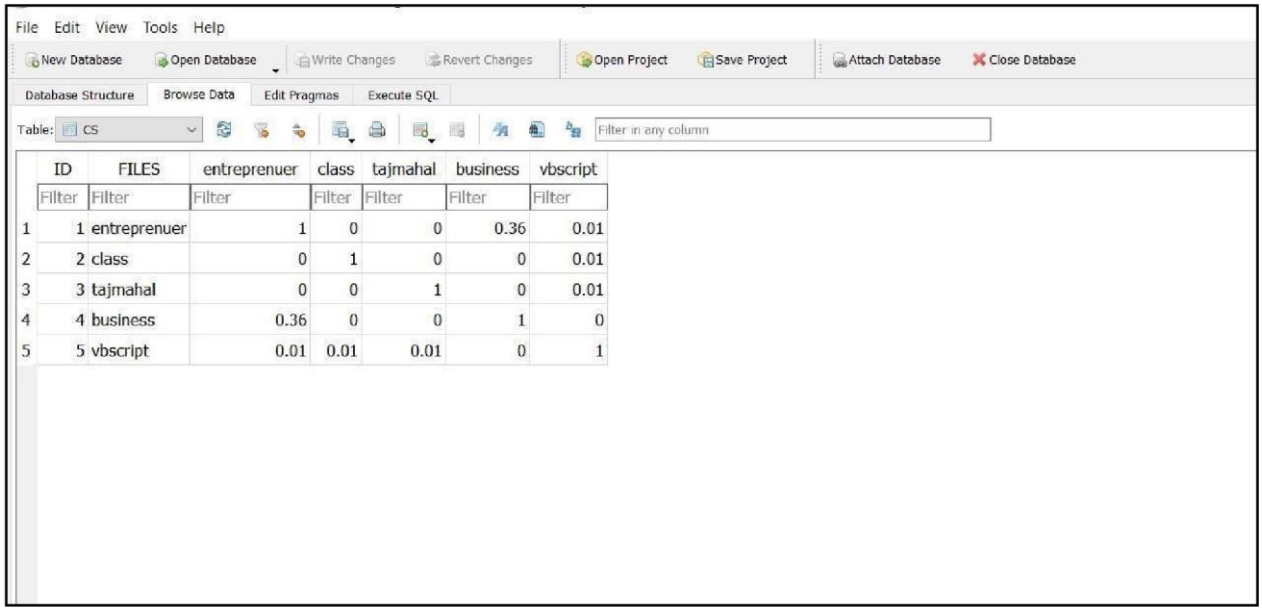
	ID	WORDS	LOG_IDF	IDF
1	1	entrepreneur	log(5/2)	0.916290731874155
2	2	defined	log(5/2)	0.916290731874155
3	3	personal	log(5/2)	0.916290731874155
4	4	risk	log(5/2)	0.916290731874155
5	5	take	log(5/2)	0.916290731874155
6	6	pursuit	log(5/2)	0.916290731874155
7	7	new	log(5/2)	0.916290731874155
8	8	business	log(5/2)	0.916290731874155
9	9	innovation	log(5/2)	0.916290731874155
10	10	form	log(5/2)	0.916290731874155
11	11	enterprise	log(5/2)	0.916290731874155
12	12	exchange	log(5/2)	0.916290731874155
13	13	taking	log(5/2)	0.916290731874155
14	14	often	log(5/3)	0.510825623765991
15	15	profit	log(5/2)	0.916290731874155
16	16	significantly	log(5/2)	0.916290731874155
17	17	success	log(5/2)	0.916290731874155
18	18	debate	log(5/1)	1.6094379124341
19	19	exact	log(5/1)	1.6094379124341
20	20	definition	log(5/1)	1.6094379124341
21	21	wide	log(5/1)	1.6094379124341
22	22	includes	log(5/1)	1.6094379124341
23	23	anyone	log(5/1)	1.6094379124341

4.7.3.4 TF-IDF Multiplier table:

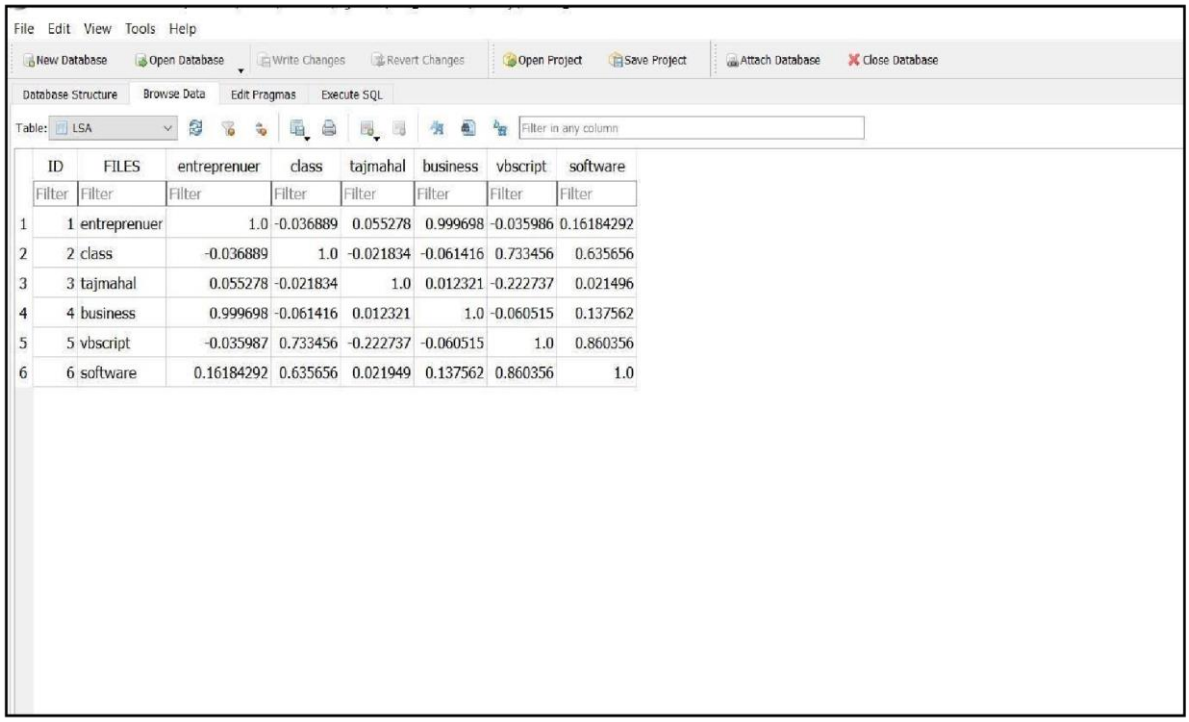


4.7.4. Cosine_Similarity Database:

4.7.4.1 CS:



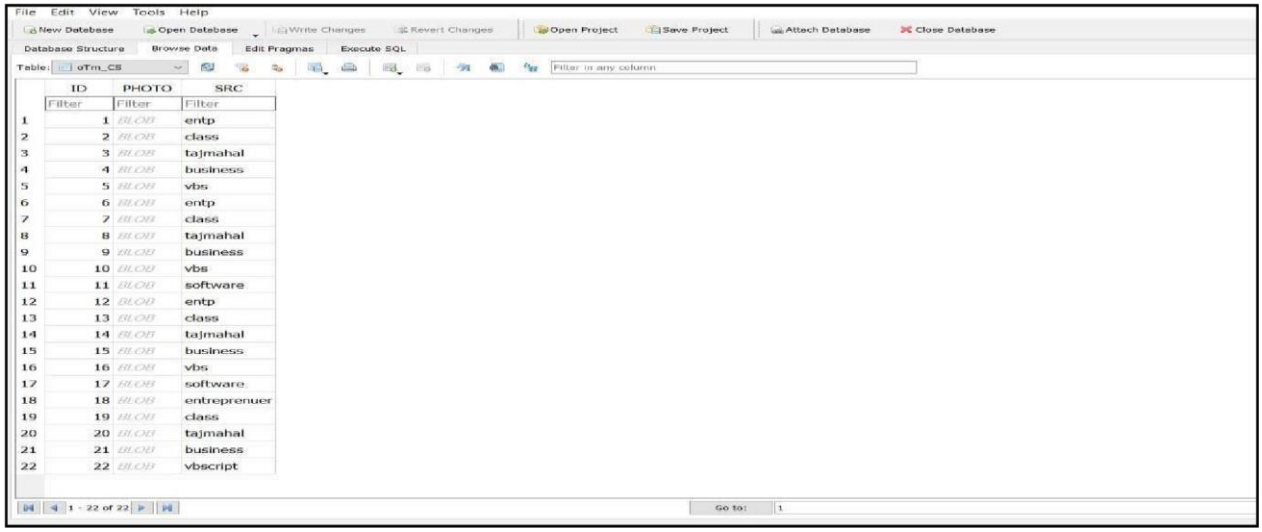
4.7.4.2 LSA:



	ID	FILES	entreprenuer	class	tajmahal	business	vbscript	software
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	entreprenuer	1.0	-0.036889	0.055278	0.999698	-0.035986	0.16184292
2	2	class	-0.036889	1.0	-0.021834	-0.061416	0.733456	0.635656
3	3	tajmahal	0.055278	-0.021834	1.0	0.012321	-0.222737	0.021496
4	4	business	0.999698	-0.061416	0.012321	1.0	-0.060515	0.137562
5	5	vbscript	-0.035987	0.733456	-0.222737	-0.060515	1.0	0.860356
6	6	software	0.16184292	0.635656	0.021949	0.137562	0.860356	1.0

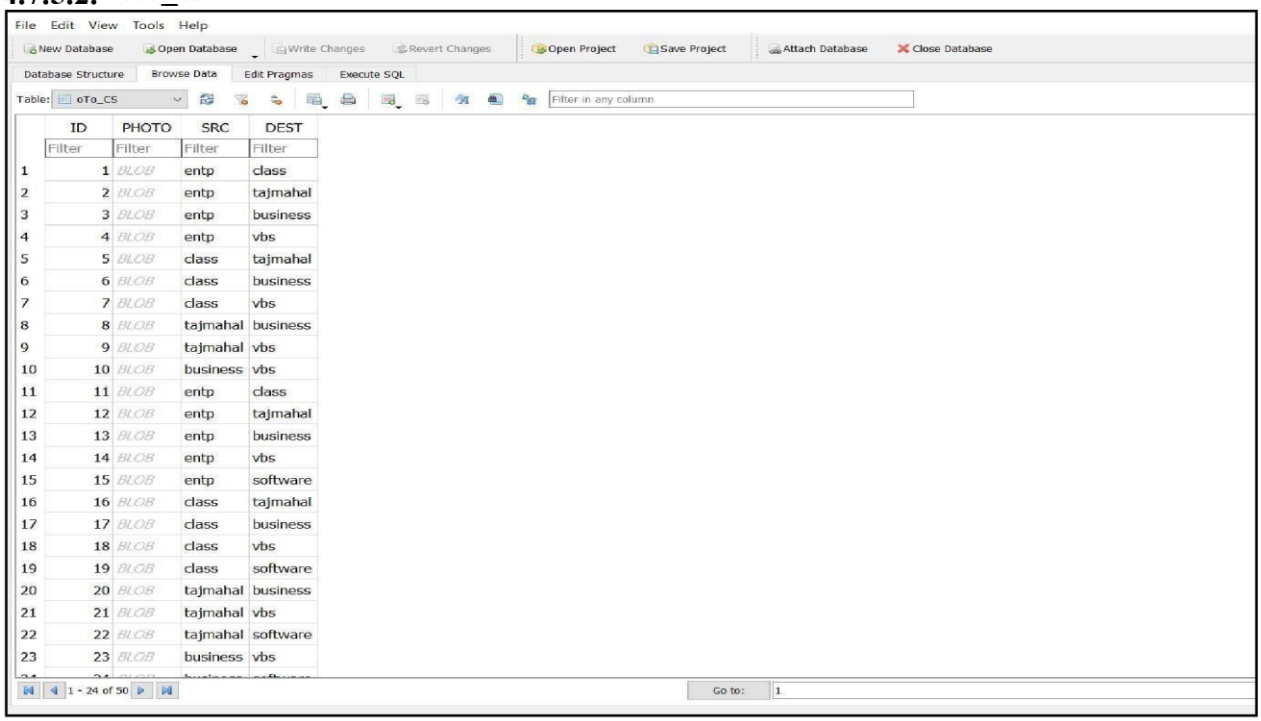
4.7.5 Graph:

4.7.5.1 otm_cs:



	ID	PHOTO	SRC
	Filter	Filter	Filter
1	1	BLOB	entp
2	2	BLOB	class
3	3	BLOB	tajmahal
4	4	BLOB	business
5	5	BLOB	vbs
6	6	BLOB	entp
7	7	BLOB	class
8	8	BLOB	tajmahal
9	9	BLOB	business
10	10	BLOB	vbs
11	11	BLOB	software
12	12	BLOB	entp
13	13	BLOB	class
14	14	BLOB	tajmahal
15	15	BLOB	business
16	16	BLOB	vbs
17	17	BLOB	software
18	18	BLOB	entreprenuer
19	19	BLOB	class
20	20	BLOB	tajmahal
21	21	BLOB	business
22	22	BLOB	vbscript

4.7.5.2. oto_cs:



File Edit View Tools Help

New Database | Open Database | Write Changes | Revert Changes | Open Project | Save Project | Attach Database | Close Database

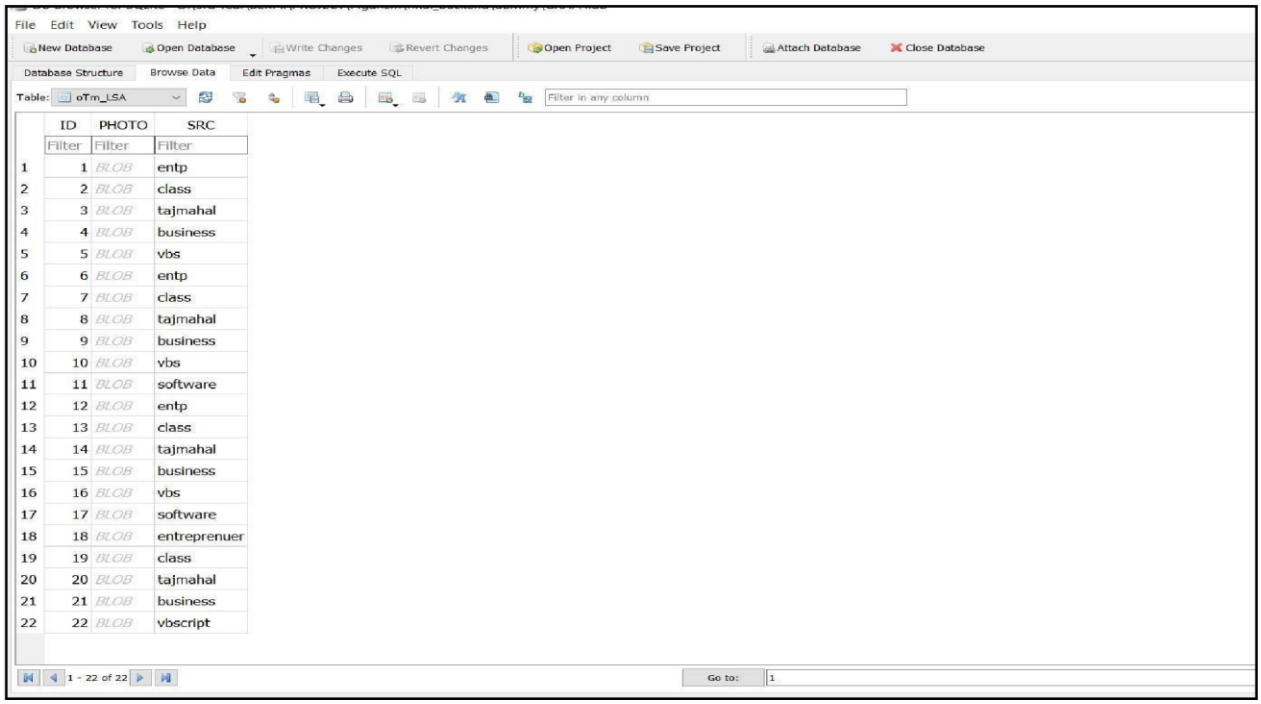
Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: oTo_CS — Filter in any column

	ID	PHOTO	SRC	DEST
	Filter	Filter	Filter	Filter
1	1	BLOB	entp	class
2	2	BLOB	entp	tajmahal
3	3	BLOB	entp	business
4	4	BLOB	entp	vbs
5	5	BLOB	class	tajmahal
6	6	BLOB	class	business
7	7	BLOB	class	vbs
8	8	BLOB	tajmahal	business
9	9	BLOB	tajmahal	vbs
10	10	BLOB	business	vbs
11	11	BLOB	entp	class
12	12	BLOB	entp	tajmahal
13	13	BLOB	entp	business
14	14	BLOB	entp	vbs
15	15	BLOB	entp	software
16	16	BLOB	class	tajmahal
17	17	BLOB	class	business
18	18	BLOB	class	vbs
19	19	BLOB	class	software
20	20	BLOB	tajmahal	business
21	21	BLOB	tajmahal	vbs
22	22	BLOB	tajmahal	software
23	23	BLOB	business	vbs

1 - 24 of 50 | Go to: 1

4.7.5.3. otm_lsa:



File Edit View Tools Help

New Database | Open Database | Write Changes | Revert Changes | Open Project | Save Project | Attach Database | Close Database

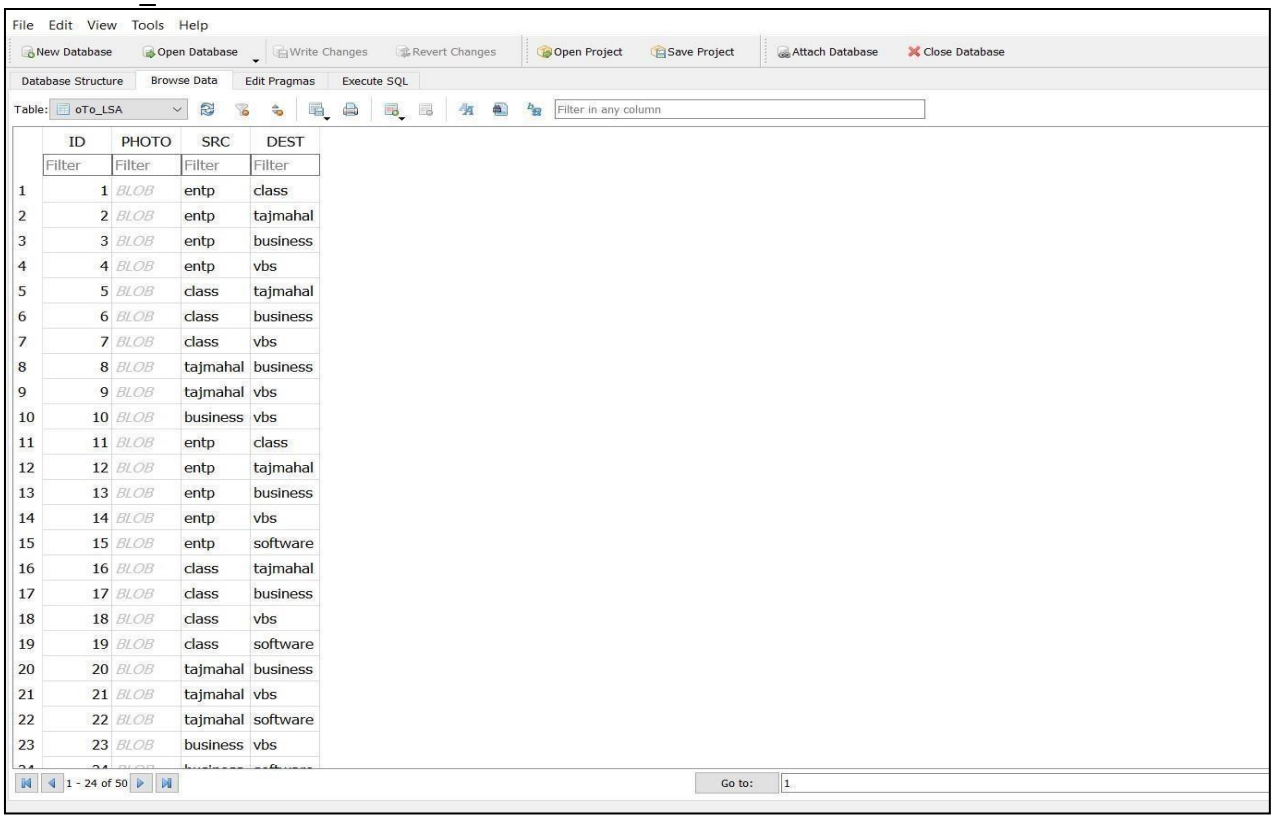
Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: oTm_LSA — Filter in any column

	ID	PHOTO	SRC
	Filter	Filter	Filter
1	1	BLOB	entp
2	2	BLOB	class
3	3	BLOB	tajmahal
4	4	BLOB	business
5	5	BLOB	vbs
6	6	BLOB	entp
7	7	BLOB	class
8	8	BLOB	tajmahal
9	9	BLOB	business
10	10	BLOB	vbs
11	11	BLOB	software
12	12	BLOB	entp
13	13	BLOB	class
14	14	BLOB	tajmahal
15	15	BLOB	business
16	16	BLOB	vbs
17	17	BLOB	software
18	18	BLOB	entreprenuer
19	19	BLOB	class
20	20	BLOB	tajmahal
21	21	BLOB	business
22	22	BLOB	vbscript

1 - 22 of 22 | Go to: 1

4.7.5.4 oto_lsa:

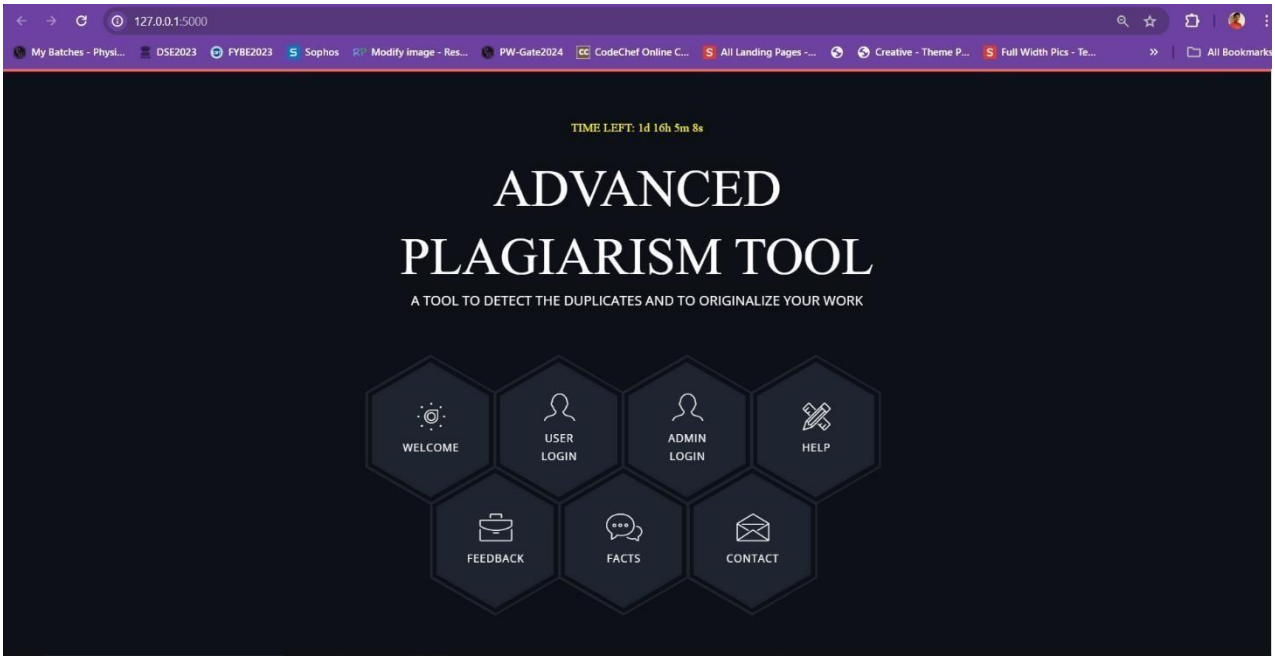


	ID	PHOTO	SRC	DEST
	Filter	Filter	Filter	Filter
1	1	BLOB	entp	class
2	2	BLOB	entp	tajmahal
3	3	BLOB	entp	business
4	4	BLOB	entp	vbs
5	5	BLOB	class	tajmahal
6	6	BLOB	class	business
7	7	BLOB	class	vbs
8	8	BLOB	tajmahal	business
9	9	BLOB	tajmahal	vbs
10	10	BLOB	business	vbs
11	11	BLOB	entp	class
12	12	BLOB	entp	tajmahal
13	13	BLOB	entp	business
14	14	BLOB	entp	vbs
15	15	BLOB	entp	software
16	16	BLOB	class	tajmahal
17	17	BLOB	class	business
18	18	BLOB	class	vbs
19	19	BLOB	class	software
20	20	BLOB	tajmahal	business
21	21	BLOB	tajmahal	vbs
22	22	BLOB	tajmahal	software
23	23	BLOB	business	vbs

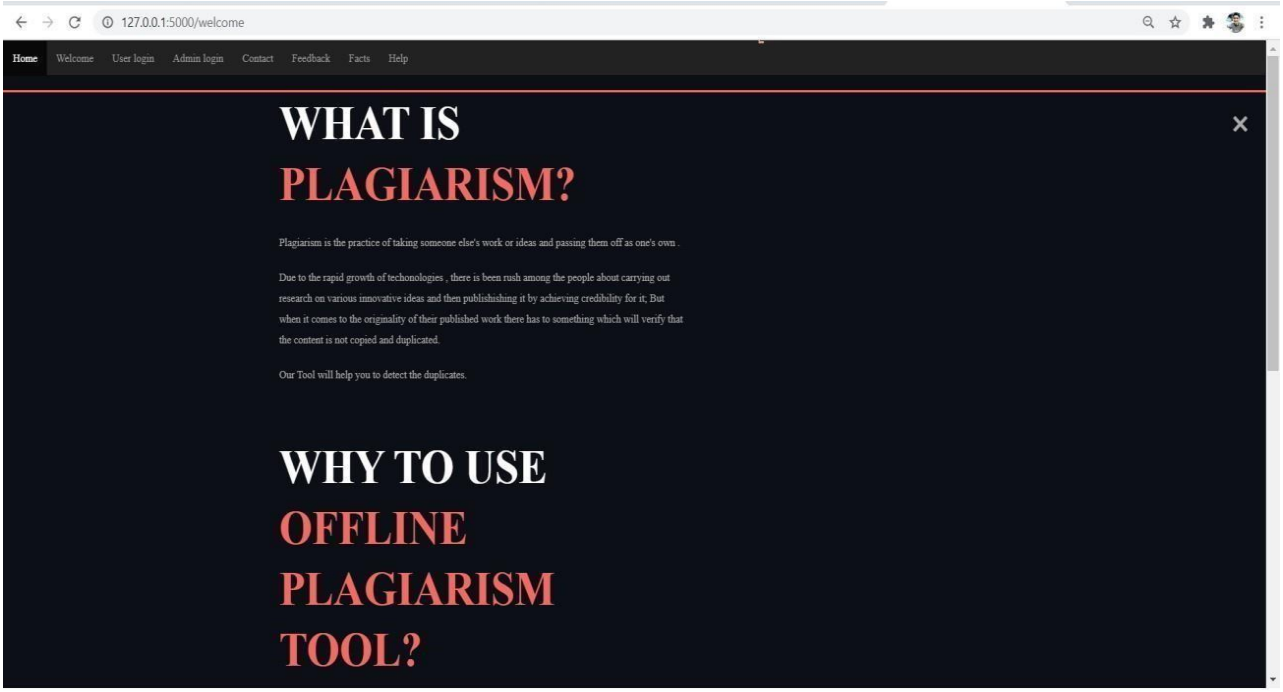
CHAPTER 05 PERFORMANCE AND RESULTS

5.1 Results:

5.1.1. Home Page:

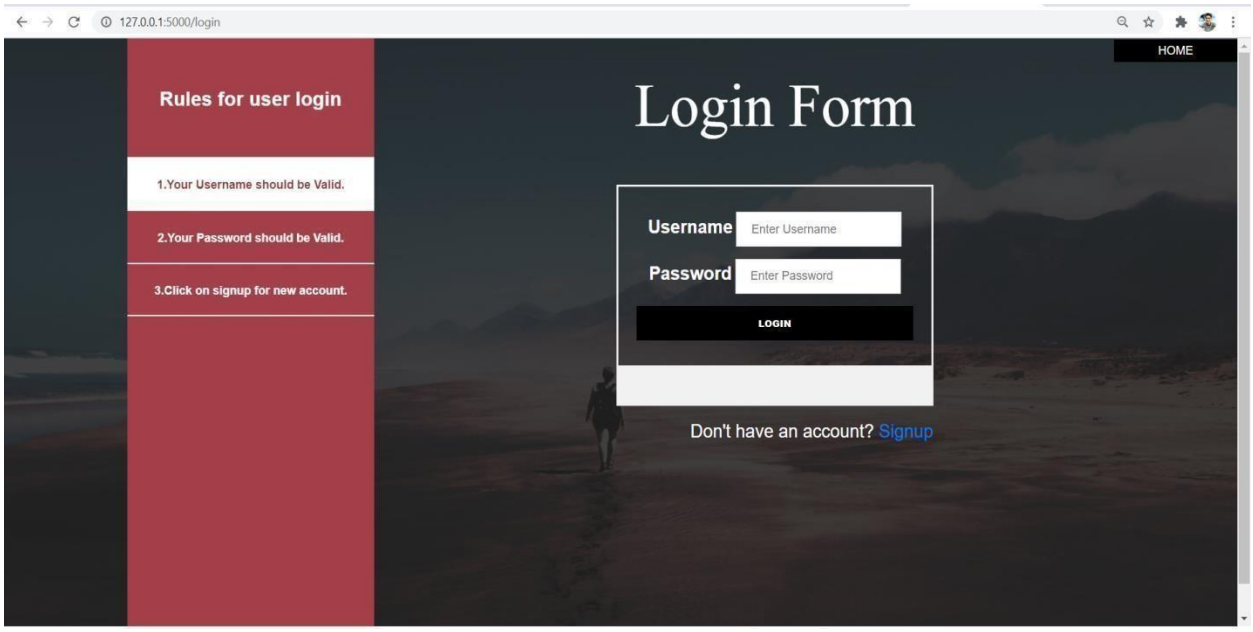


5.1.2. Welcome Page:

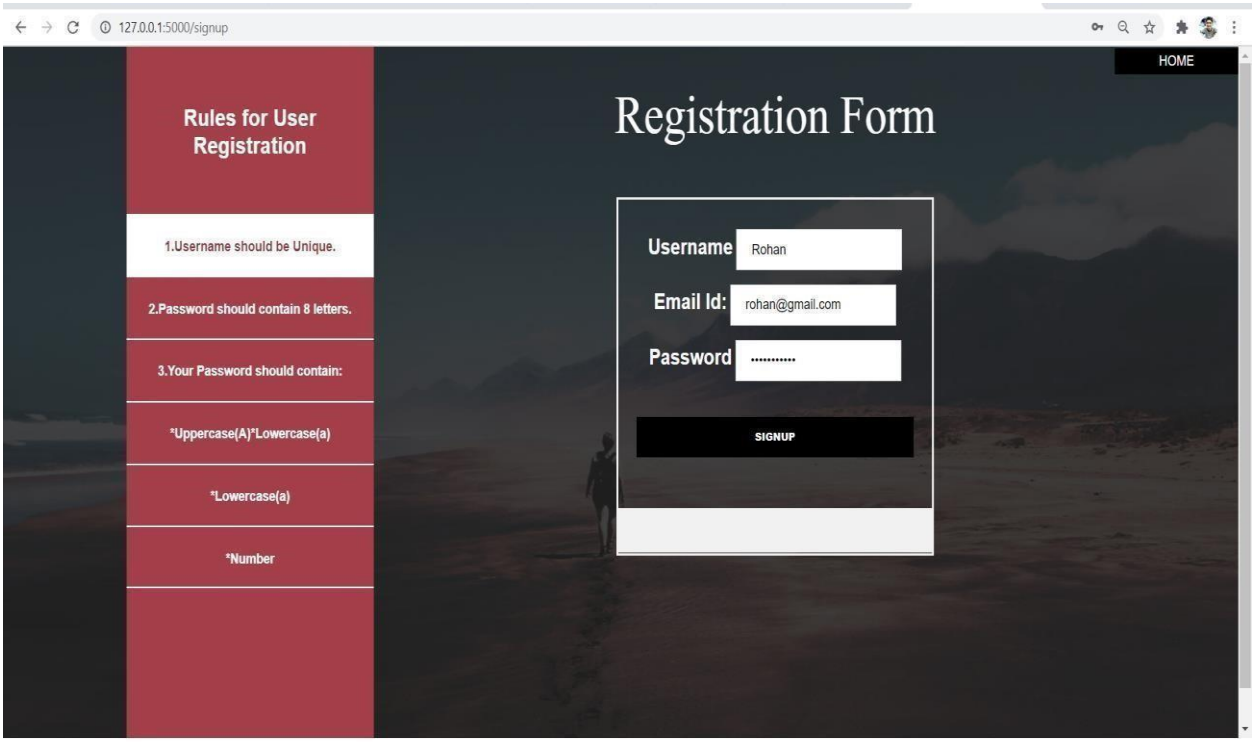


5.2 User Login (Before Deadline):

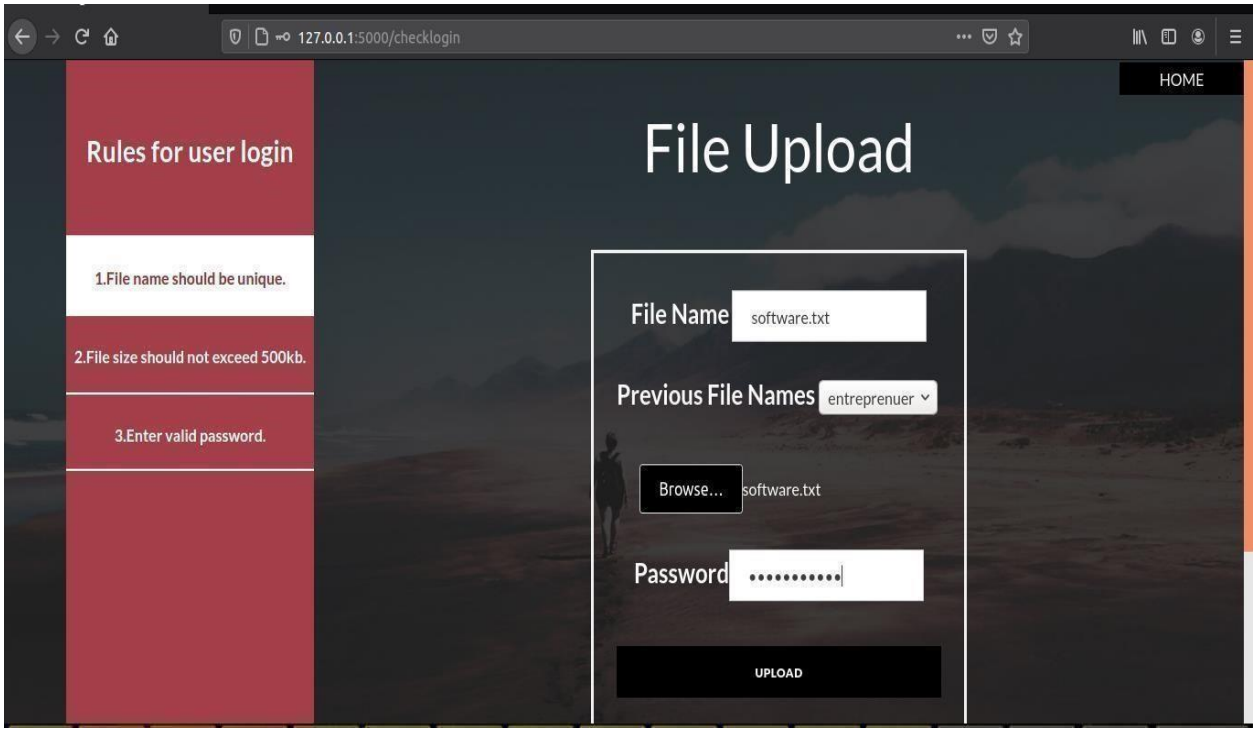
5.2.1 Login page:

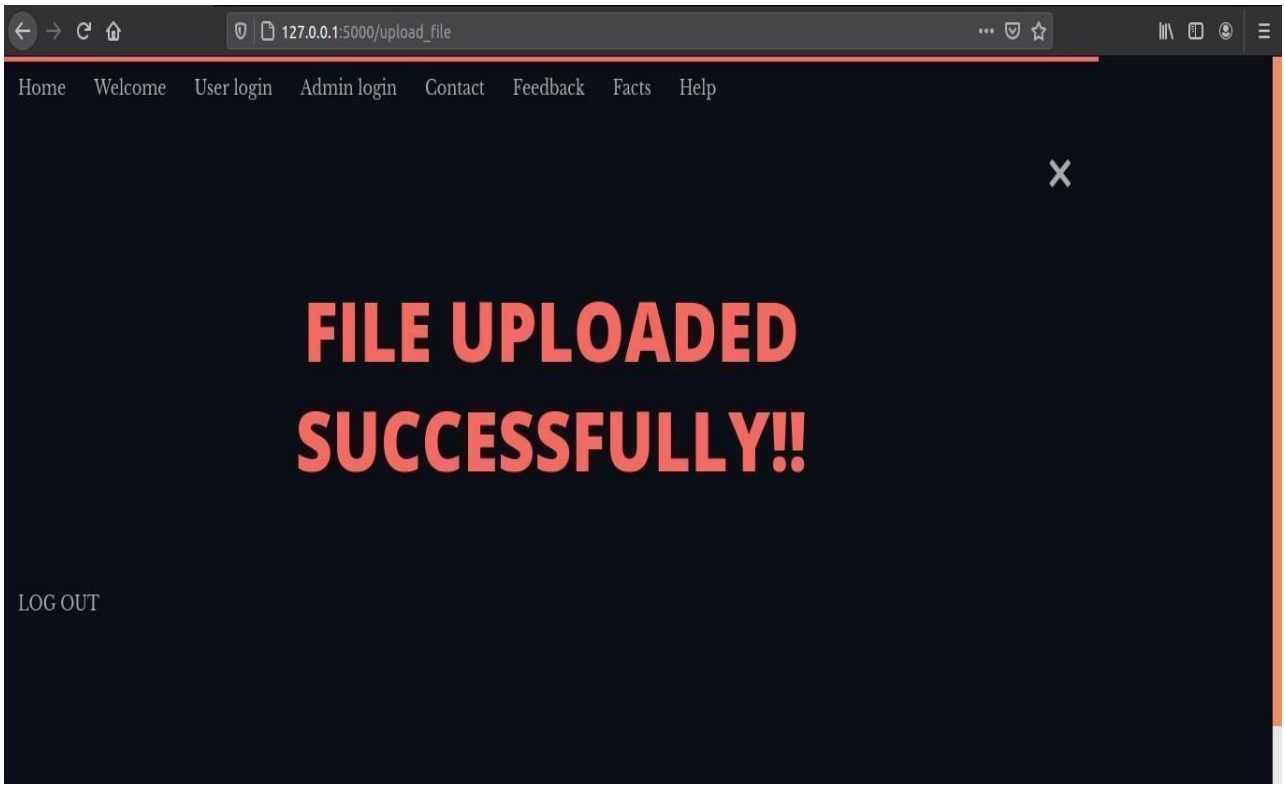


5.2.2 Signup Page:



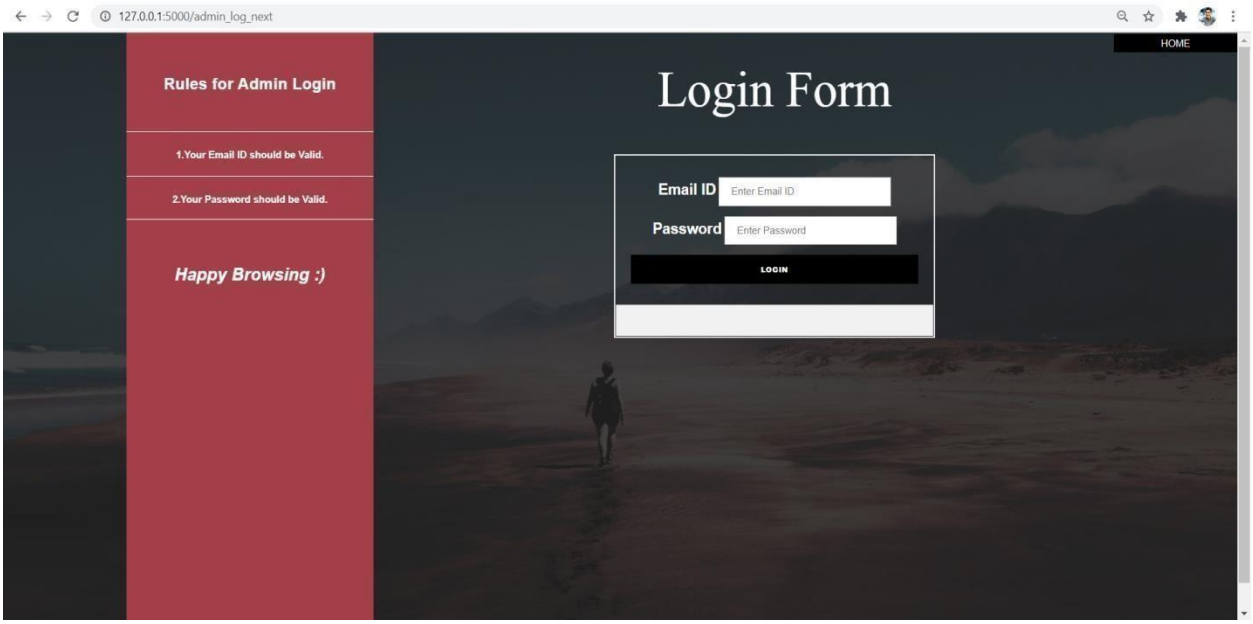
5.2.3 File Upload Page:



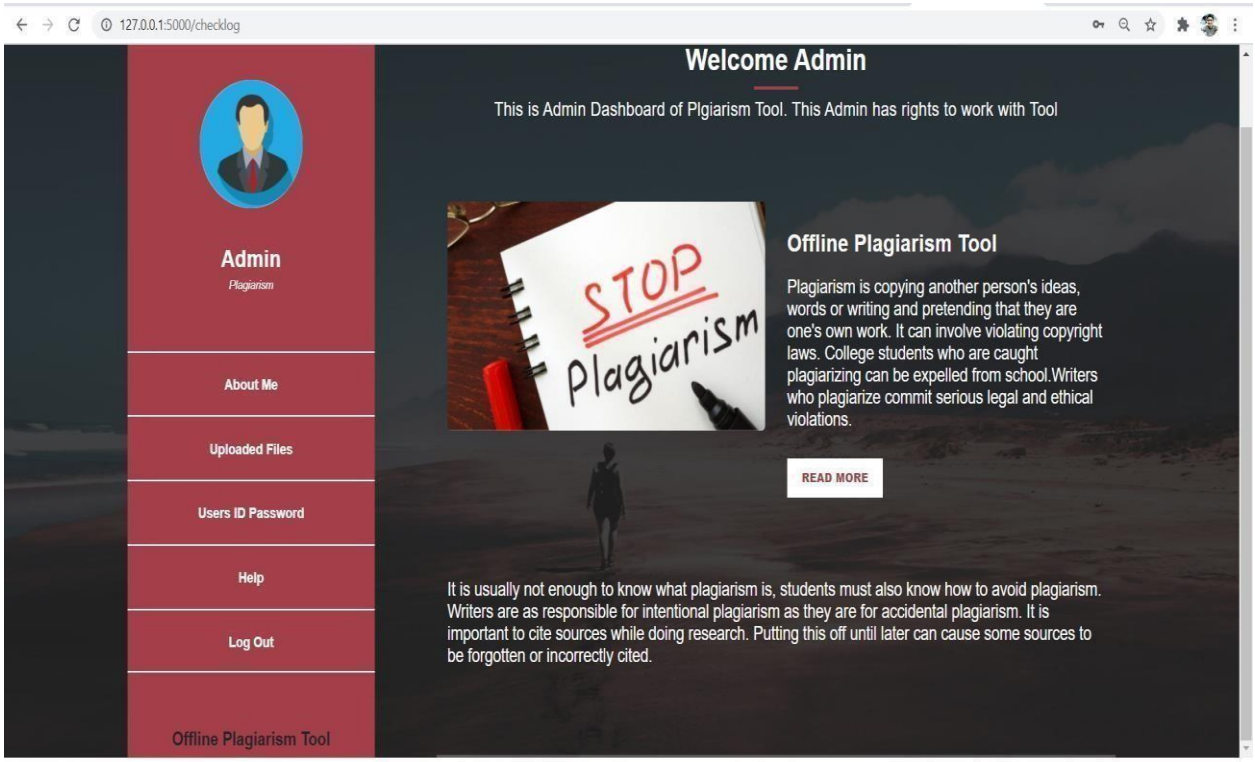


5.3 Admin Login (Before Deadline):

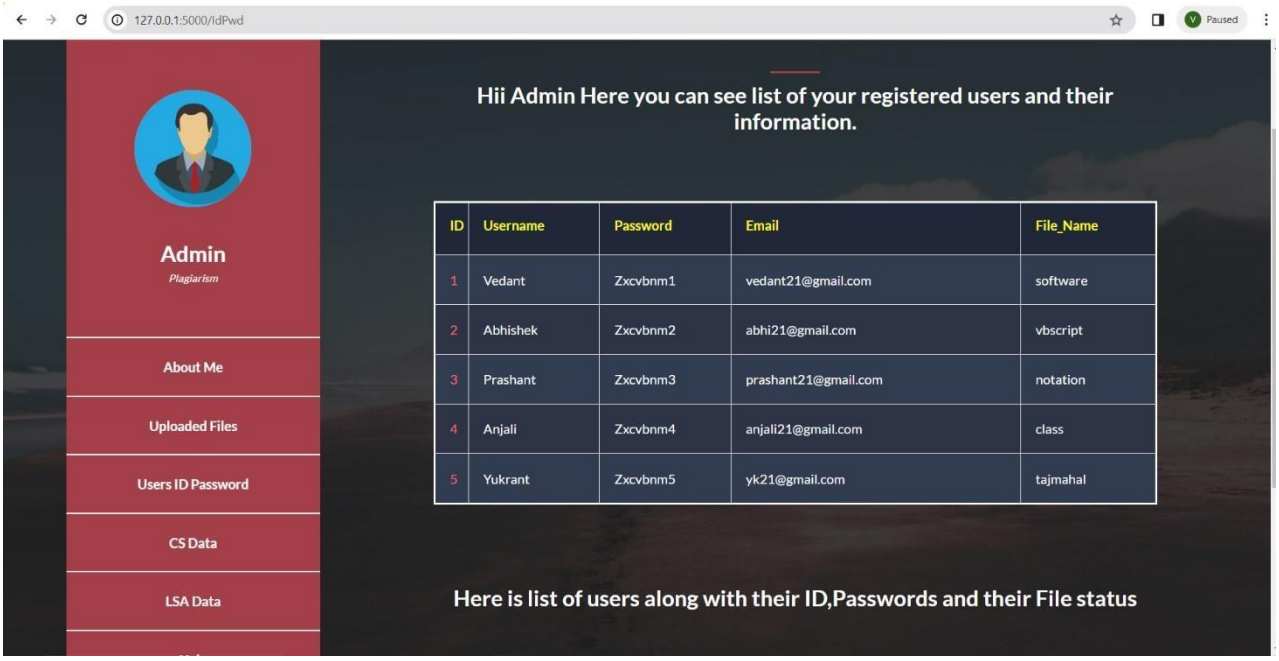
5.3.1 Login Page:



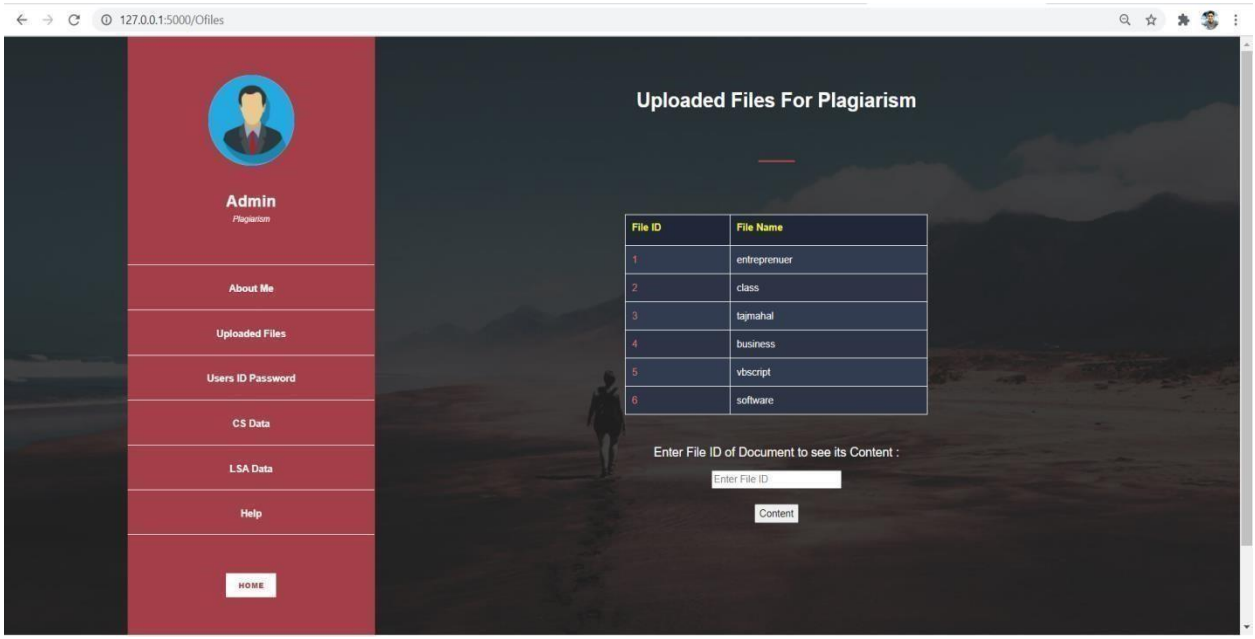
5.3.2 Admin Dashboard:

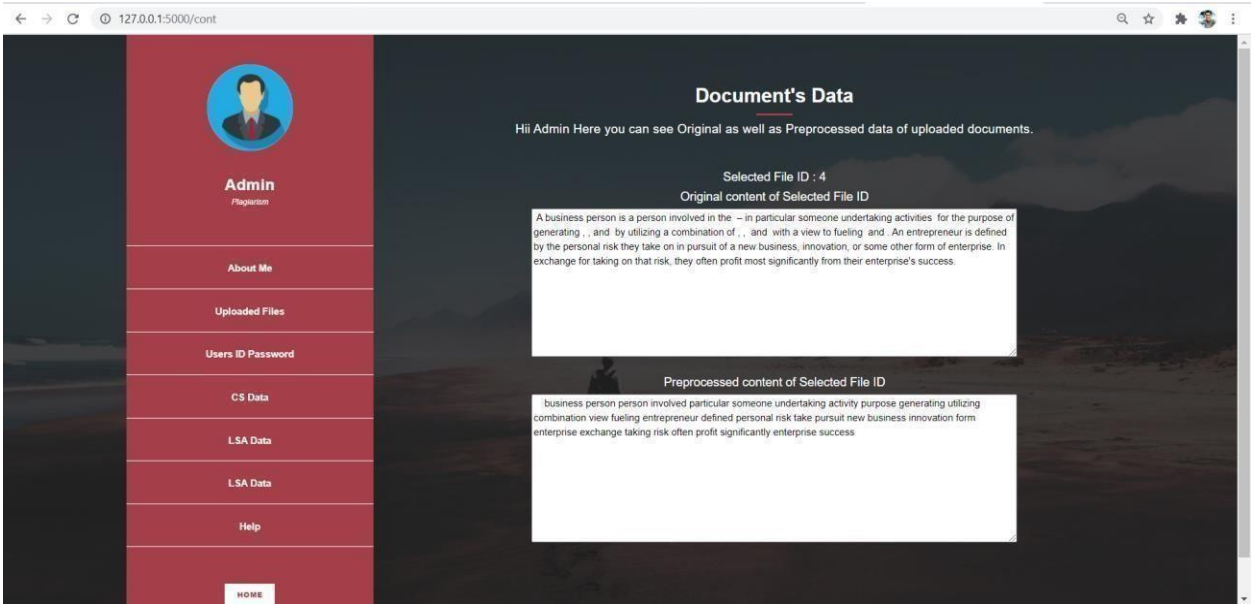


5.3.3 User details:



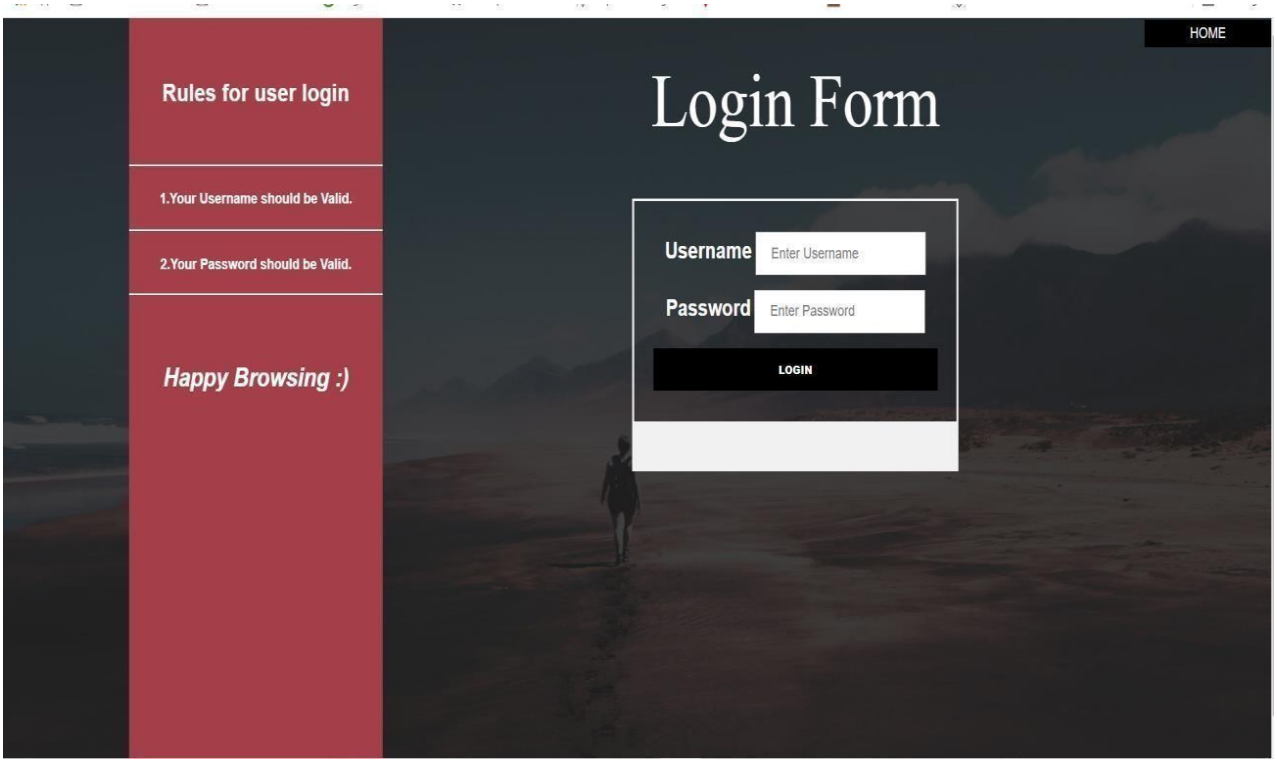
5.3.4 Uploaded files:



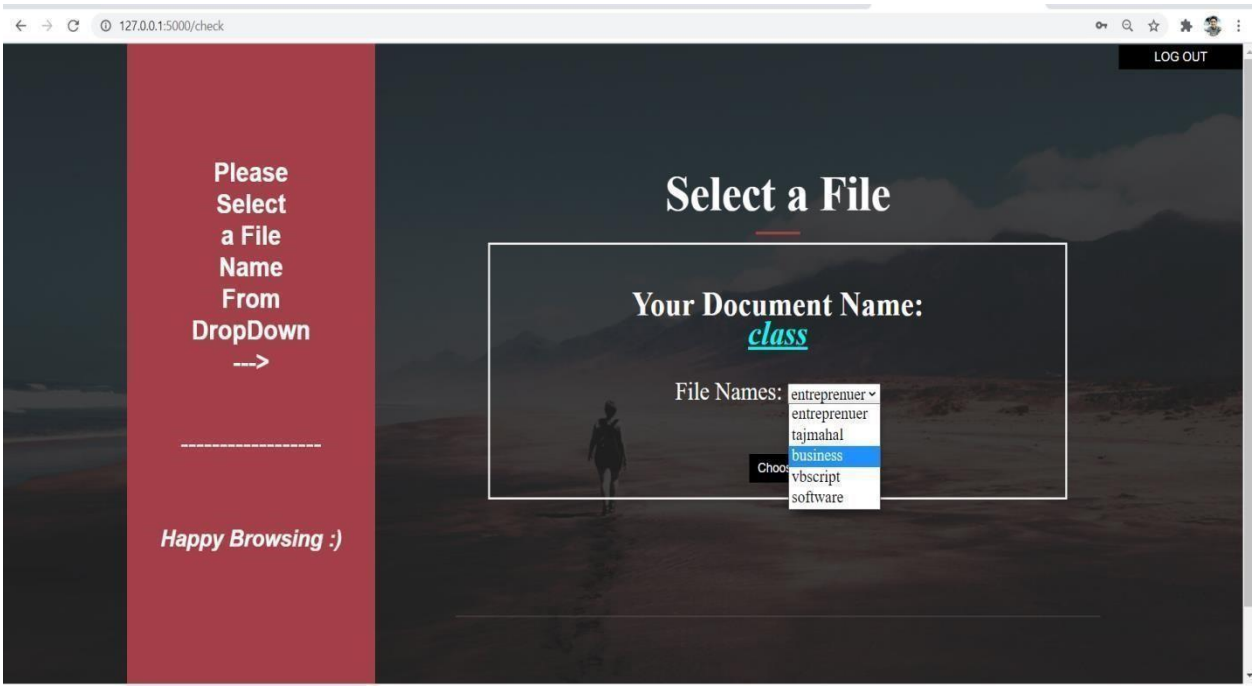


5.4 User login (After Deadline):

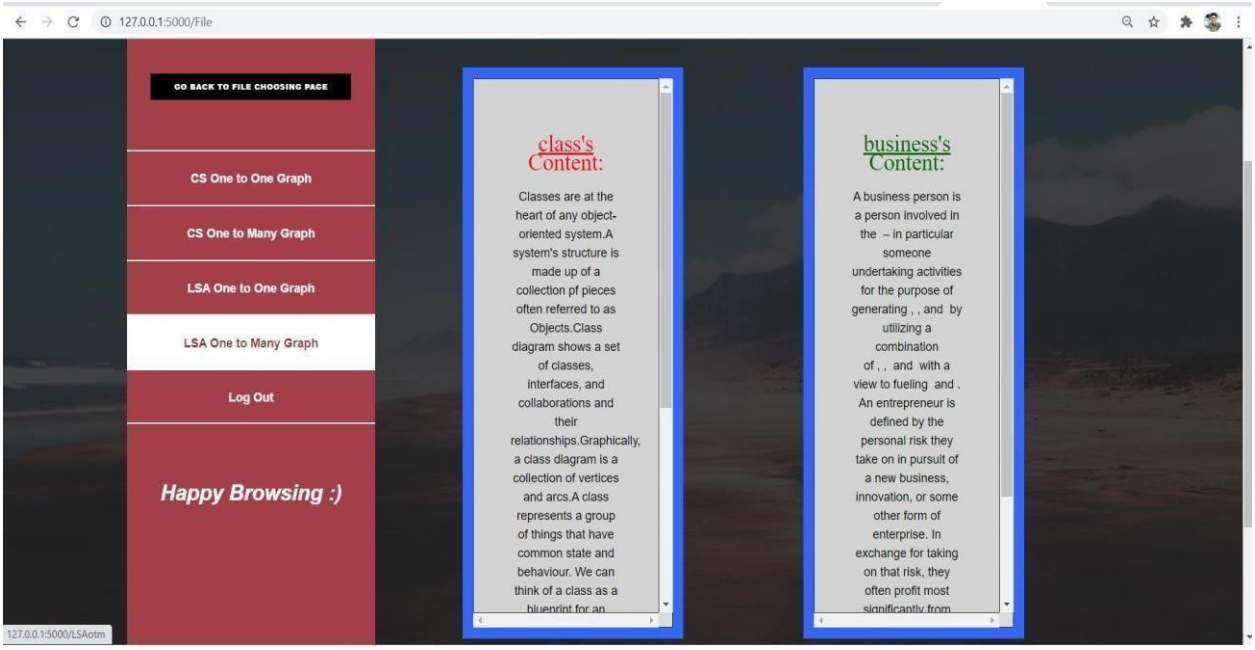
5.4.1 Login page:



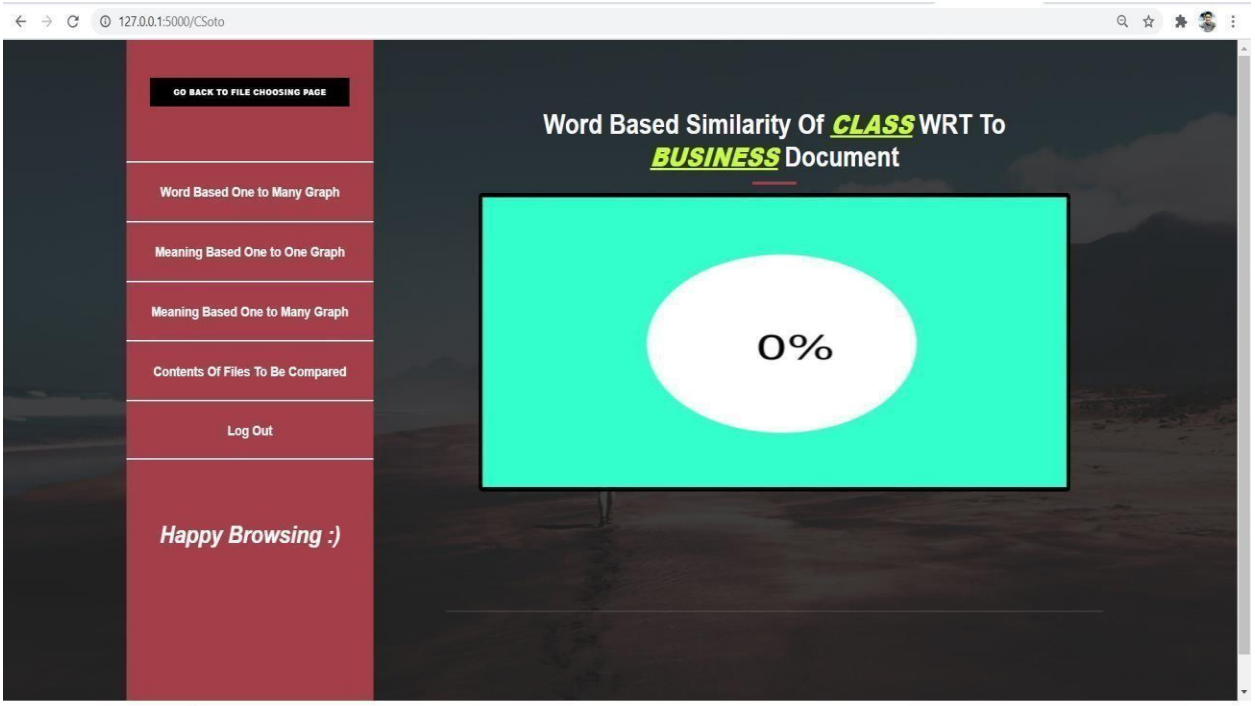
5.4.2 File selection page:



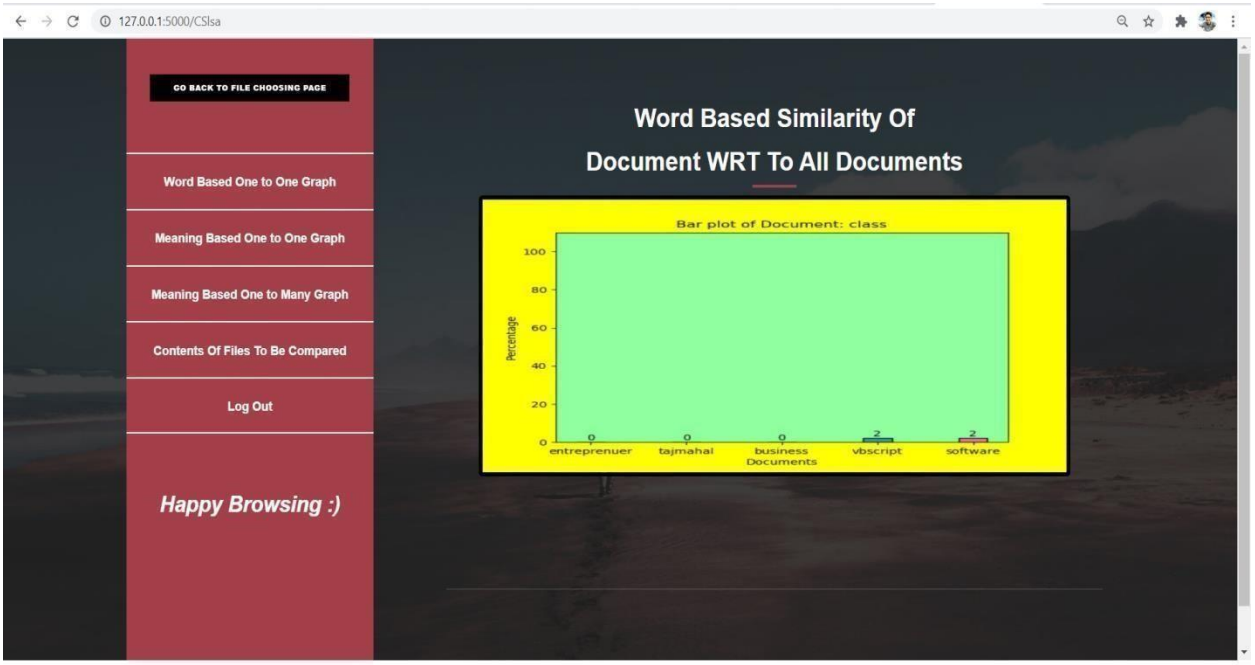
5.4.3 Contents of documents:



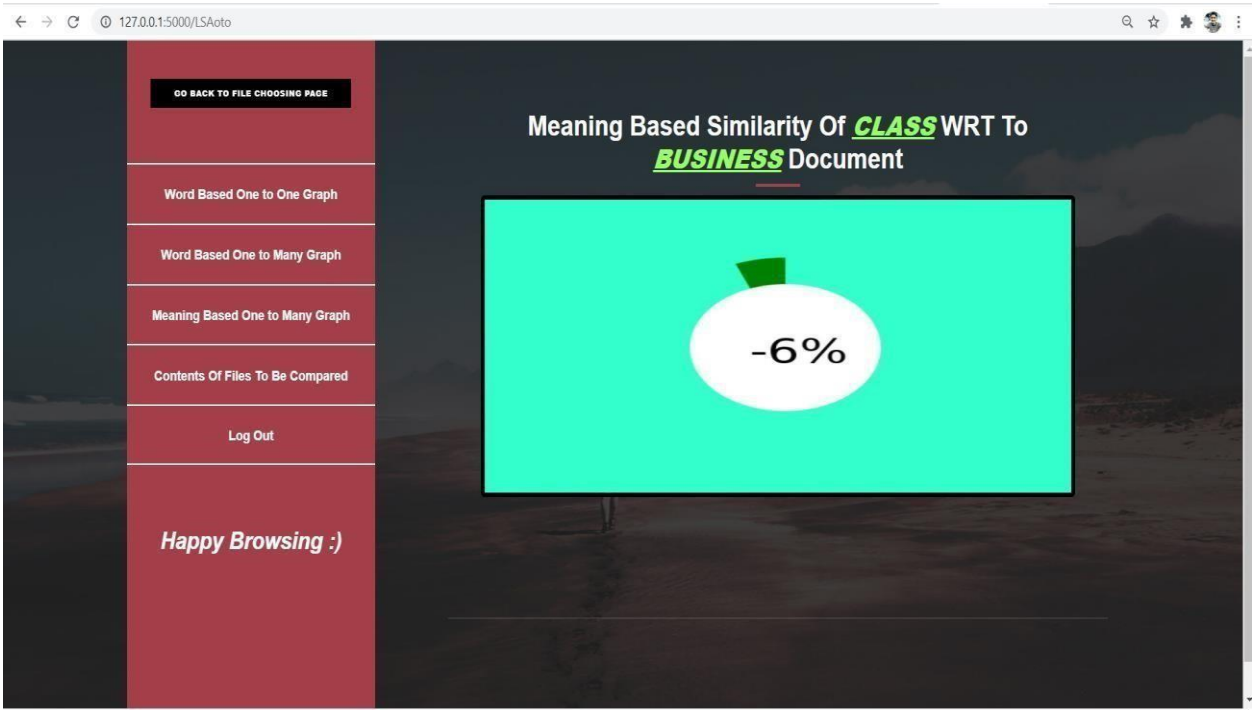
5.4.4 Word Based Similarity of one document with respected to one document:



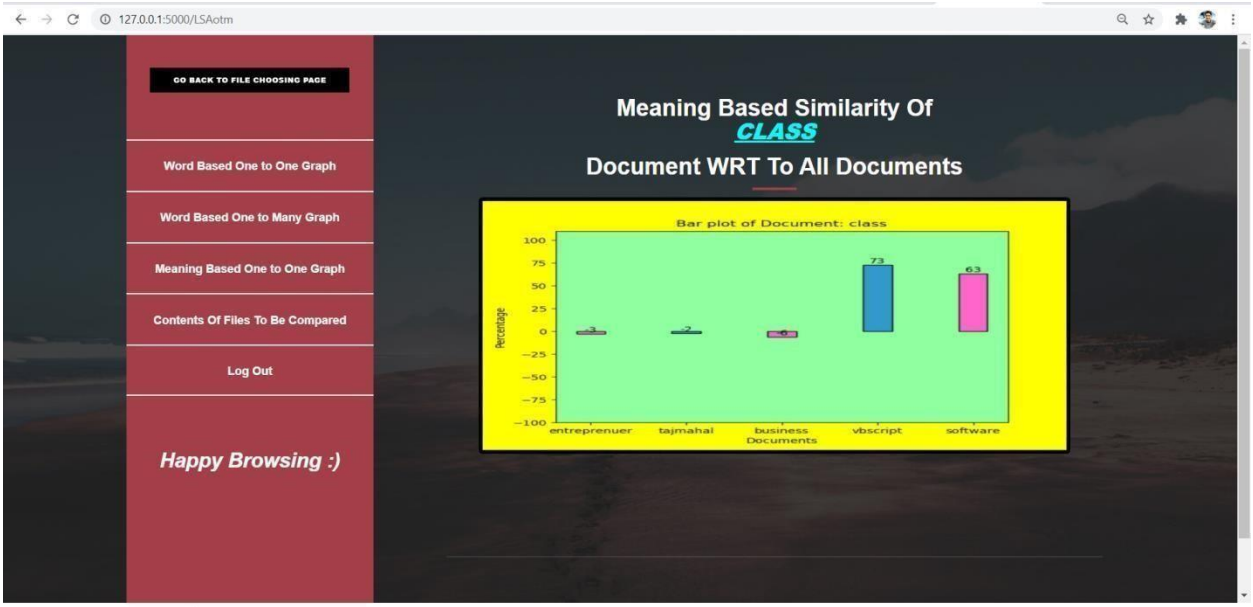
5.4.5 Word Based Similarity of one document with respected to all documents:



5.4.6 Meaning Based Similarity of one document with respected to one document:

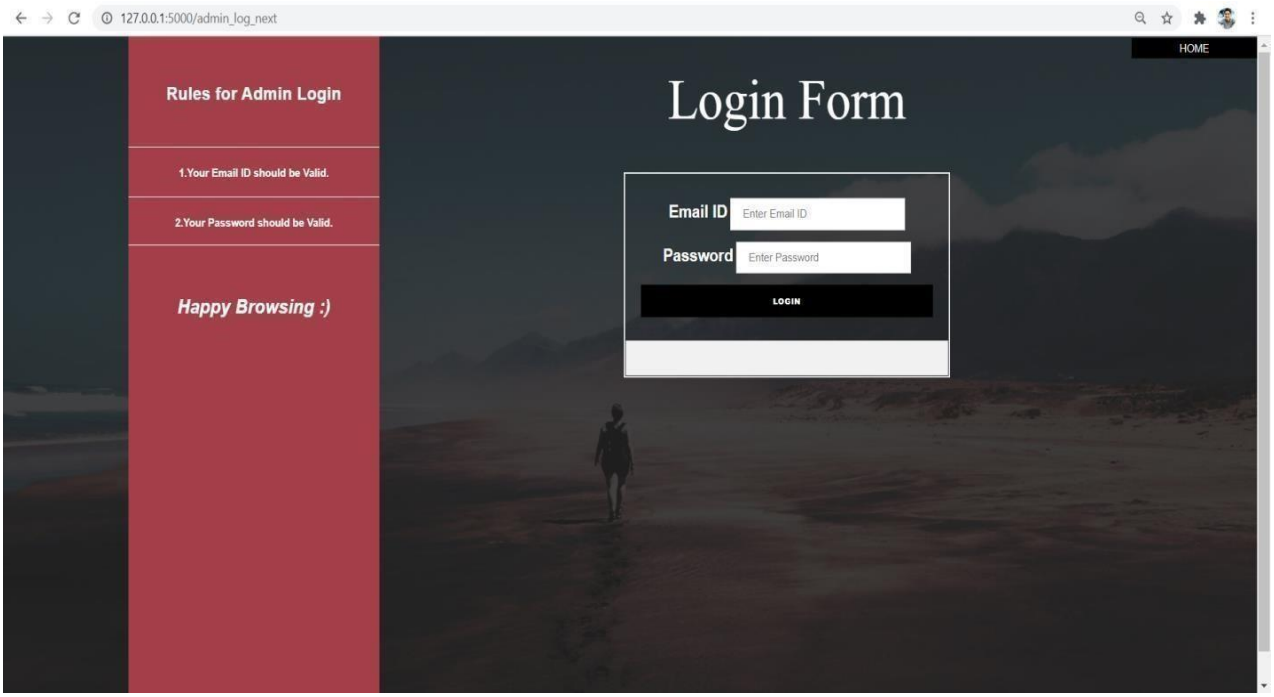


5.4.7 Meaning Based Similarity of one document with respected to all documents:

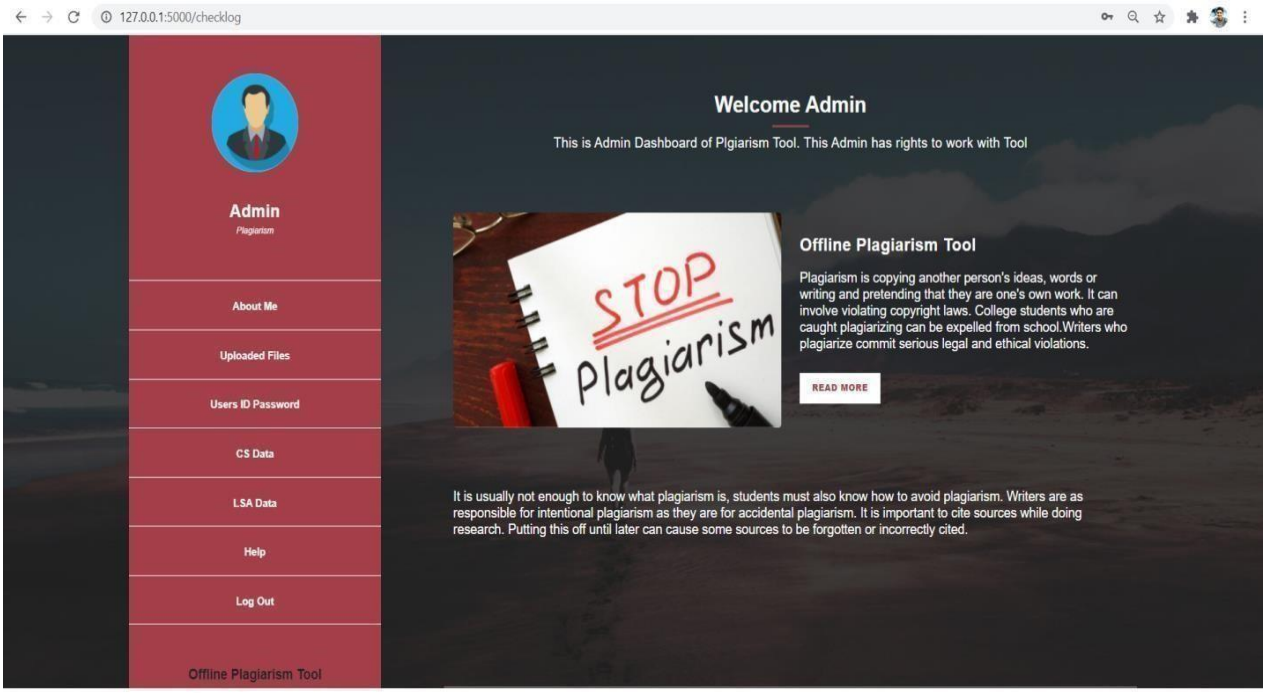


5.5 Admin Login (After deadline):

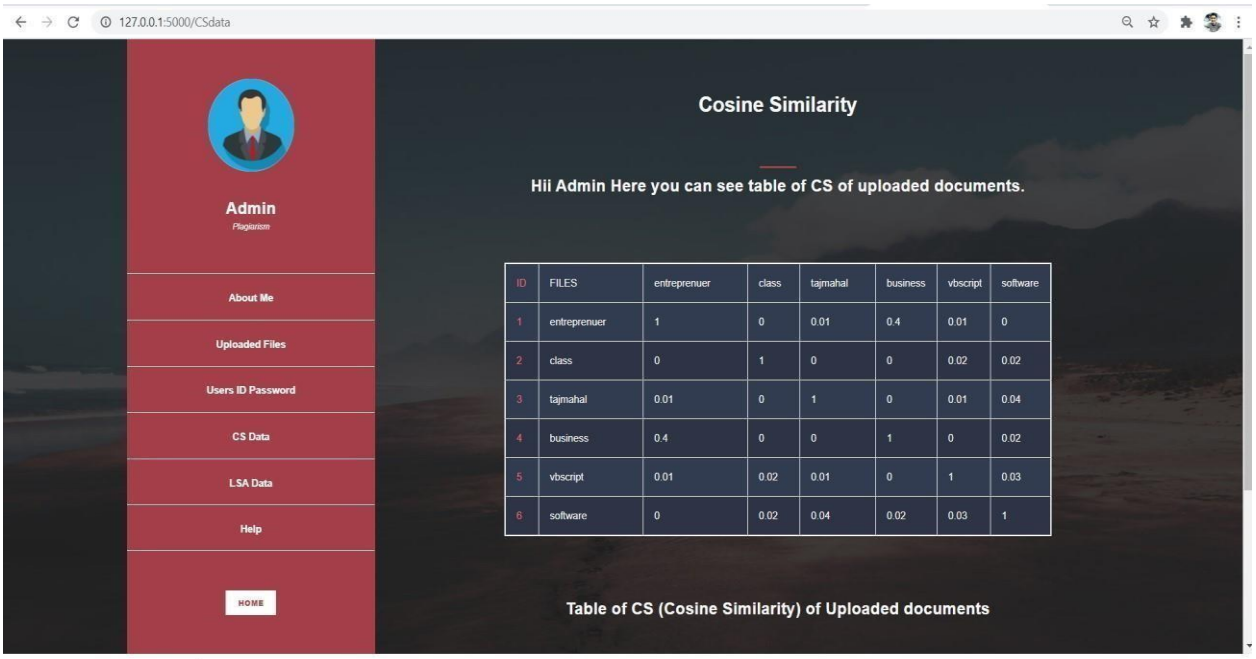
5.5.1 Login page:



5.5.2 Admin Dashboard:

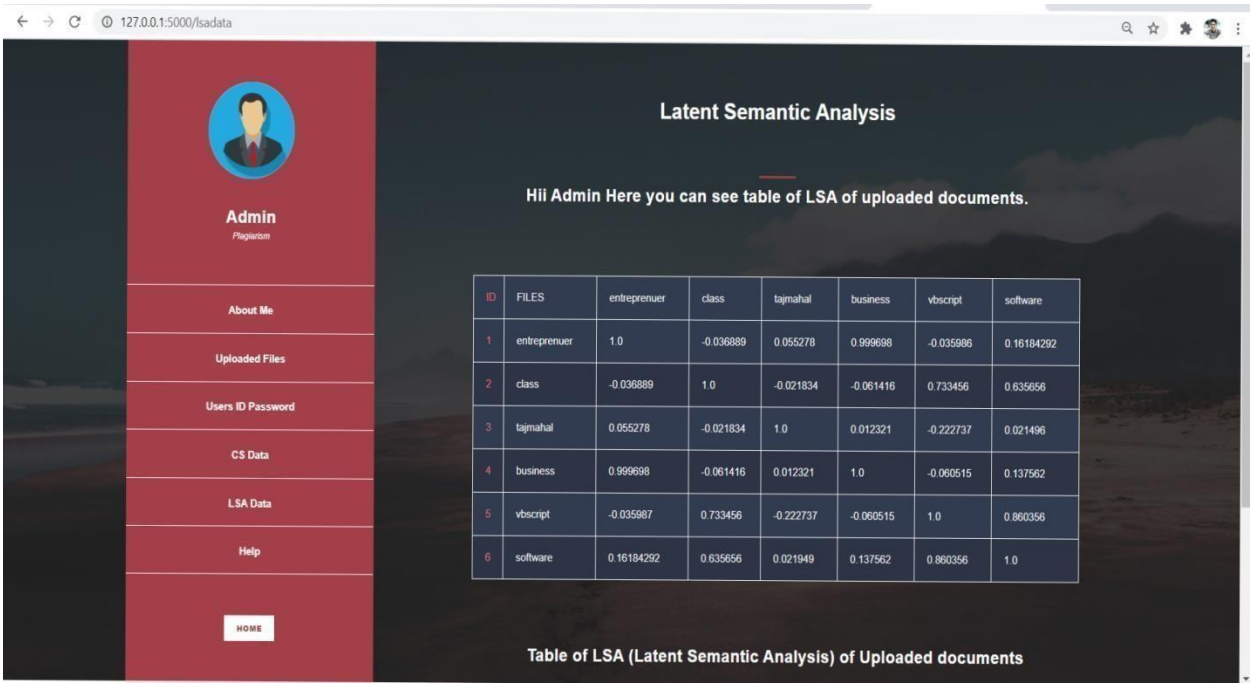


5.5.3 CS data:



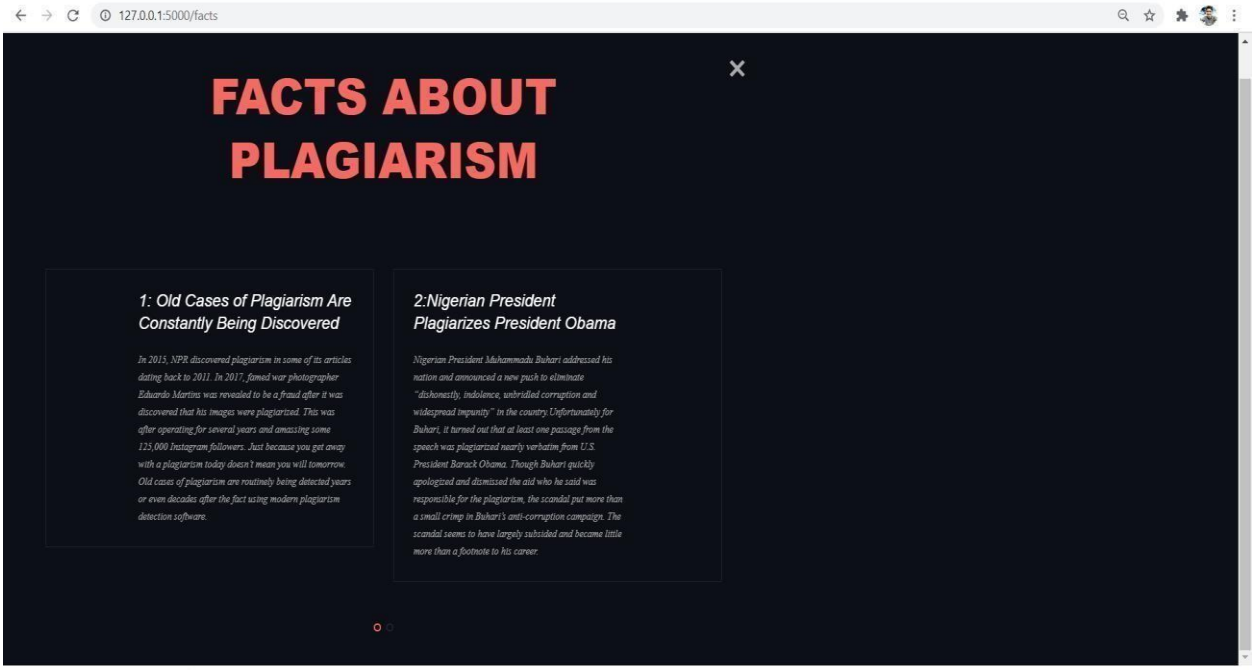
ID	FILES	entreprenuer	class	tajmahal	business	vbscript	software
1	entreprenuer	1	0	0.01	0.4	0.01	0
2	class	0	1	0	0	0.02	0.02
3	tajmahal	0.01	0	1	0	0.01	0.04
4	business	0.4	0	0	1	0	0.02
5	vbscript	0.01	0.02	0.01	0	1	0.03
6	software	0	0.02	0.04	0.02	0.03	1

5.5.4 LSA data:



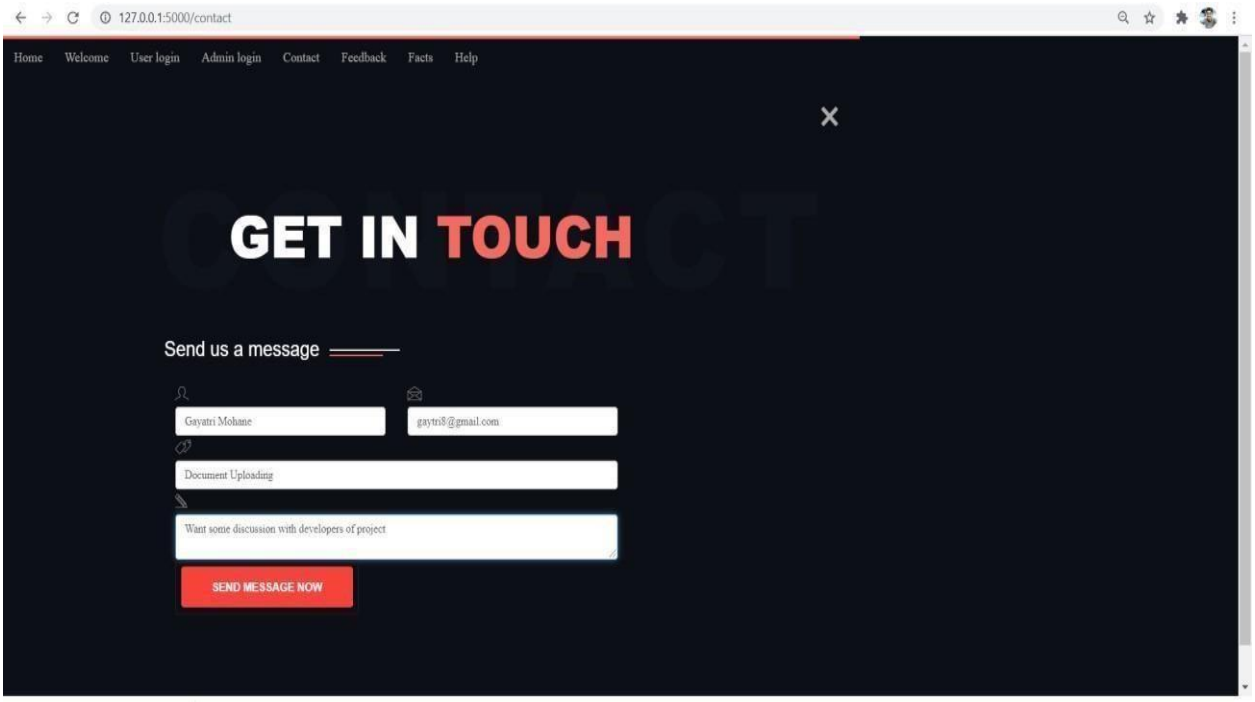
ID	FILES	entreprenuer	class	tajmahal	business	vbscript	software
1	entreprenuer	1.0	-0.036889	0.055278	0.999698	-0.035986	0.16184292
2	class	-0.036889	1.0	-0.021834	-0.061416	0.733456	0.635656
3	tajmahal	0.055278	-0.021834	1.0	0.012321	-0.222737	0.021496
4	business	0.999698	-0.061416	0.012321	1.0	-0.060515	0.137562
5	vbscript	-0.035987	0.733456	-0.222737	-0.060515	1.0	0.860356
6	software	0.16184292	0.635656	0.021949	0.137562	0.860356	1.0

5.5.5 Facts about Plagiarism:

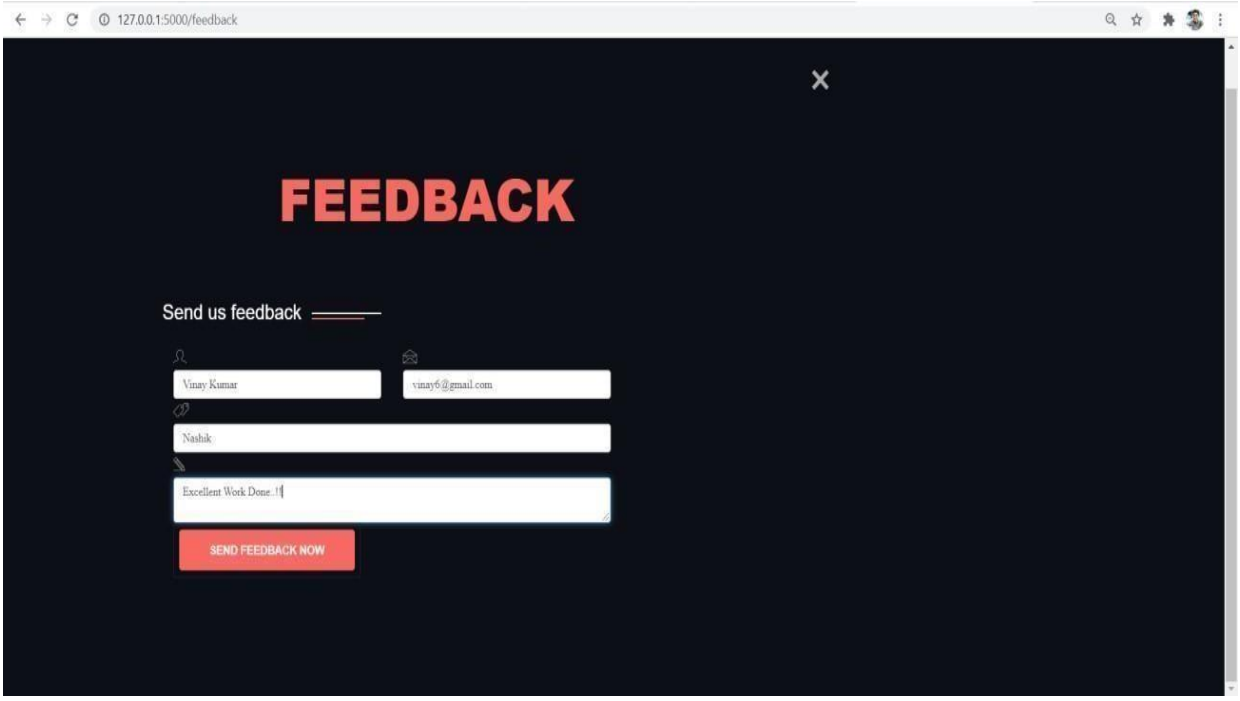


65

5.5.7 Contacts Page:



5.5.8 Feedback Page:



CHAPTER 06 COSTING

6.1. Cost of Project:

Sr. No.	Title	Cost
1	Work hour	80(per person)
2	Group members	05
3	Charges per hour	400(per person)
4	Total work charges	80*400=32000(per person)
5	Total charges	32000* 5=160000
6	Internet hours	100
7	Cost per hour	50
8	Total internet charges	100*50=5000
9	Printing and other costs	350*5=1750
10	Computer charges	2000
11	Total Cost	Rs. 1642500

CHAPTER 07 CONCLUSION

7.1. Future Scope:

Though there are many such tools available but this tool can be very useful for the organization for their internal use because they can compare or use their domain concerned documents which will give them effective results.

It involves both challenges and opportunities as technology advances and societal attitudes toward intellectual property continue to evolve. will likely involve a combination of technological, educational, legal, and cultural developments aimed at promoting originality, integrity, and responsible use of information in an increasingly digital and interconnected world.

7.2. Conclusion:

The main focus of this project is to provide accurate similarity score to the user on the basis of meaning as well as on the basis of words. It also focuses on providing the results in graphical format so that the results are more readable and understandable. This system uses algorithms like Cosine Similarity and technique like Latent Semantic Analysis to accomplish the task accurately. The systems require document which is further processed and results are obtained. The system can be proved very useful to the organization for comparing their domain concerned documents which will eventually reduce redundancy within.

7.3. References:

[1] Ms. Pooja Kherwa1, Dr.Poonam Bansal :“Latent Semantic Analysis: An Approach to Understand Semantic of Text” in International Conference on Current Trends in Computer, Electrical, Electronics and Communication, (ICCTCEEC-2017)

[2] Dongyang Yan, Keeping Li, Shuang Gu, Liu Yang : “Network-Based Bag-of-Words Model for Text Classification” in IEEE Access , 2015

[3] Gleen A. Dalaorao , Ariel M.Sison , Ruji P. Medina :“Integrating Collocation as TFIDF Enhancement to Improve Classification Accuracy” in IEEE , 2019

[4] Ankur Agarwal, Christopher Baechle, Ravi Behara, Xingquan Zhu : “A Natural Language Processing Framework for Assessing Hospital Readmission.

[5] Oi Mean Foong, Suet Peng Yong , Farha Am Jaid , “Text summarization using latent semantic analysis model in mobile android platform.” , 2015 Asia Modelling Symposium.

[6] H. Shaikh and A. Kumar, "A Comparison between Syllabus of AICTE and various Universities - An NLP Based Approach," 2021 2nd International Conference for Emerging Technology (INCET), Belagavi, India

[7] S. P. and A. P. Shaji, "A Survey on Semantic Similarity," 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, India

[8] T. Islam, M. Hossain and M. F. Arefin, "Comparative Analysis of Different Text Summarization Techniques Using Enhanced Tokenization," 2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh

[9] M. Saeed and A. Y. Taqa, "An Intelligent Approach for Semantic Plagiarism Detection in Scientific Papers," 2022 8th International Conference on Contemporary Information Technology and Mathematics (ICCITM), Mosul, Iraq

[10] F. Ahmad and M. Faisal, "Comparative Study of Techniques used for Word and Sentence Similarity," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India

[11] D. Viji and S. Revathy, "Semantic Similarity Detection from text document using XLNet with a DKM-Clustered Bi-LSTM Model," 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India

[12] D. Srivamsi, O. M. Deepak, M. D. A. Praveena and A. Christy, "Cosine Similarity Based Word2Vec Model for Biomedical Data Analysis," 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India

[13] P. Kherwa and P. Bansal, "Latent Semantic Analysis: An Approach to Understand Semantic of Text," 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, India,

[14] L. Shkurti, J. Ajdari, F. Kabashi and V. Fusa, "PlagAL: Plagiarism detection system for Albanian texts," 2021 10th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro

ACKNOWLEDGEMENT

We would like to take this opportunity to sincerely thank Dr. Shilpa Kabra, our guide and assistant professor in the Department of Information Technology at the Government College of Engineering, Chhatrapati Sambhaji Nagar, for her direction and assistance during the study for our project. We could not have done this without her gracious advice and assistance. We appreciate her prompt input, which enabled us to efficiently monitor and plan the process. Her ideas, time, and support enabled us to finish our assignment quickly and effectively. We also thank Dr. Anjana Ghule for her support, for creating a wonderful learning atmosphere, and for providing the necessary resources.

We are grateful to all of my B.E. instructors as well as Dr. Anjana Ghule, Head of the Department of Information Technology at the Government College of Engineering Chhatrapati Sambhaji Nagar, for their insightful counsel. We also sincerely appreciate the participation of all of the faculty members, non-teaching personnel, and friends.

BE20F06F026 - Prashant Kadam

BE21S06F002 - Anjali Dhole

BE21S06F003 - Abhishek Joshi

BE21S05F009 - Vedant Wagh

(Department of Information Technology)

Government College of Engineering, Aurangabad (Chhatrapati SambhajiNagar)

DECLARATION

We thus certify that, with Dr. Shilpa Kabra's assistance, we developed, finished, and wrote the dissertation titled **"Advanced Plagiarism Tool".** We have followed all ethical and academic guidelines and acknowledged the usage of all materials in this endeavour. Any data, thoughts, or ideas that came from other people have been properly referenced in accordance with the guidelines.

Date:

Place: Chhatrapati SambhajiNagar

PRASHANT KADAM
(BE20F06F026)

ANJALI DHOLE
(BE21S06F002)

ABHISHEK JOSHI
(BE21S06F003)

VEDANT WAGH
(BE21S06F009)