A

Major Project

On

IMAGE PLAGIARISM

Submitted To

Jawaharlal Nehru Technological University, Hyderabad

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

B.JOHN NOVA (177R1A05C5)

B.KUMAR (177R1A0567)

M.MADHUMITHA (177R1A0588)

P.HARSHITH KUMAR (167R1A05H0)

Under the esteemed guidance of

M.MADHUSUDHAN

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGC Act.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**IMAGE PLAGIARISM**" being submitted by **B.JOHN NOVA (177R1A05C5)**,**B.KUMAR (177R1A0567), M.MADHUMITHA (177R1A0588), P.HARSHITH KUMAR (167R1A05H0)** in partial fulfillment of the requirements for the award of the degree of B. Tech in Computer Science and Engineering of the Jawaharlal Nehru Technological University Hyderabad, is a record of Bonafide work carried out by him/her under our guidance and supervision during the year 2021-22.

The result embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

M.MADHUSUDHAN **DR. A. Raji Reddy**

Assistant professor **Director**

Internal Guide

Dr. K. Srujan Raju **EXTERNAL EXAMINER**
HOD

Submitted on viva voice Examination held on_____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide

M.MADHUSUDHAN , Assistant. Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Mr. A. Uday Kiran, Mr. J. NarasimhaRao, Dr. T. S. Mastan Rao, Mrs. G. Latha ,Mr. A. Kiran Kumar** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to the Head of the Department **Dr. K. Srujan Raju** for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

We are obliged to our Director **Dr. A. Raji Reddy** for being cooperative throughout the course of this project. We would like to express our sincere gratitude to our Chairman Sri. **Ch. Gopal Reddy** for his encouragement throughout the course of this project

The guidance and support received from all the members of **CMR TECHNICAL CAMPUS** who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement without which this assignment would not be possible. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project.

B.JOHN NOVA (177R1A05C5)

B.KUMAR (177R1A0567)

M.MADHUMITHA (177R1A0588)

P.HARSHITH KUMAR (167R1A05H0)

ABSTRACT

In an educational environment, plagiarism is a crucial task that needs to be identified, in recent years all known journals and conferences, as well as universities, request a plagiarism report from students and researchers to prove the originality of published text or scientific paper. Plagiarism detection usually checks the text content via many of the platforms which are available for productive use reliably identifying copied text or near-copies of text and these systems usually fail to detect the images, and File's plagiarism since it is originally built for text mainly. In this paper, we suggest an adaptive, scalable, and extensible, robust method for image plagiarism.

Image plagiarism is stealing of another's work and passing it off as their own work without crediting the source. Image plagiarism is based on image processing, which helps to manipulate and perform operations on image to detect plagiarism. Previously lot of work is done to detect plagiarism on text, but there is no much work done in this area. In this paper an attempt is made to detect difference between the images using image subtraction. The system is also overcoming the vulnerability of re-sizing, compression and color differentiation. The similarity and the difference between the images are displayed using histogram.

LIST OF FIGURES

LIST OF SCREENSHOTS

1.INTRODUCTION

INTRODUCTION

1.1 PROJECT SCOPE

Now-a-days with humungous usage of internet there are more chances of plagiarism in different fields in different ways. The rapid development of information technology, especially the Internet is pointed out to be factor driving the student to practice plagiarism. Prevent efforts continue to be made both from government policies and stakeholder by creating software anti plagiarism. However, in the reality the practice of plagiarism remains common and relatively more widespread. Therefore, the practice of plagiarism should not only be viewed from the perspective of the academic violations, but also from the other perspectives such as detecting copyrights , detecting plagiarised documents etc.

1.2 PROJECT PURPOSE

The objectives of any plagiarism checker include to find the similarities in the text and ensure that the document is original. It further implies that no part of the document is copied from other writer’s work. A plagiarism checker is a nice way for checking the originality of a document. The objective of our application not only includes text plagiarism but also has image plagiarism detection with which image similarity can be found.

1.3 PROJECT FEATURES

Plagiarism is the representation of another author's language, thoughts, ideas, or expressions as one's own original work. A plagiarized document can be made by copying content from set of source files. Plagiarism is considered a violation of academic integrity and a breach of journalistic ethics . Generally, plagiarism is not in itself a crime, but like counterfeiting fraud can be punished in a court for prejudices caused by copyright infringement, violation of moral rights, or torts.

How to Avoid Plagiarism?

- Cite Your Source.
- Present your own idea.

2.SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

2.1 EXISTING SYSTEM:

There are many tools available in the web, exact algorithms of many commercial tools are not known, whereas the general approaches for existing plagiarism detection researches are mainly non-NLP based.

These methods included:

• Relative frequency models

• Dot plot visualisation of matching sequences of words on charts

• Document fingerprinting using frequency-based strategies

Over the years many methodologies have been developed to perform automatic detection of plagiarism, including tools for natural language text detection such as

i) MOSS(Aiken,1994)

ii) ii) Turtin (iParadigms, 2010)

iii) iii)CopyCatch (CFL software, 2010)

In detecting image plagiarism The existing methodology maybe sufficient for detecting plagiarism of images when the source and suspected image have not been rotated by a large margin, but in case of rotational changes the existing methodology will fail.

DISADVANTAGES

This system has the vulnerability of re-sizing, compression and colour differentiation. CBIR requires lot of work to detect whether the images are plagiarized or not.

The existing system is not efficient to detect plagiarism properly for different types of images.

2.2 Proposed System

In the proposed system it uses five modulus method(FMM)

Since each entry in the size table needs 0(1) time to compute, and because there are two nested loops in the algorithm -(LCS)-, where one of them is looping n times and the inner is looping m times, then total running time of the algorithm is O(mn). The space of this problem is O(mn) as well, but it can be reduced considerably to O(n) if the trace back of longest subsequence is not necessarily needed [1].

The proposed work mainly focus on finding the similarity between two images. Sample image is given as the reference and it is compared with the other image which is taken from any journal and comparison is done through histogram. Histogram is the best way to visualize the largest intensities of an image. It is used to find the problems which originate during image acquisition such as exposure, contrast etc.

Advantages of proposed system:

In the proposed system the image result is displayed using histogram.It is the best way to

Visualize the largest intensities of an image.

This system also overcomes the vulnerability of re-sizing,compression,colour differentiation.

The main advantage of image compression this will reduce the original size of image to lower size.so,this makes the image processing faster.

This system also capable to detyect plagiarism in UML diagrams,flow charts,architecture and even in snap shot of test results.

2.3 SYSTEM REQUIREMENTS;

HARDWARE REQUIREMENTS:

- System Processor : Min i3 and above
- Hard Disk : min 120GB
- RAM : 4GB or above
- System type : 64 bit operating system
- Input Devices : Keyboard,Mouse

SOFTWARE REQUIREMENTS:

●Operating system **:**Windows 8/10 or Linux

●Tools :Django,Python shell

●Software : Python 3.7

2.4 FUNCTIONAL REQUIREMENTS

1.Data Collection

2.Data Preprocessing

3.Training And Testing

4.Modiling

5.Predicting

2.5 NON FUNCTIONAL REQUIREMENTS

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *"how fast does the website load?"* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are > 10000. Description of non-functional requirements is just as critical as a functional requirement.

- Usability requirement
- Serviceability requirement
- Manageability requirement
- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement
- Scalability requirement
- Interoperability requirement
- Reliability requirement

- Maintainability requirement
- Regulatory requirement
- Environmental requirement

2.6 SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not

have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

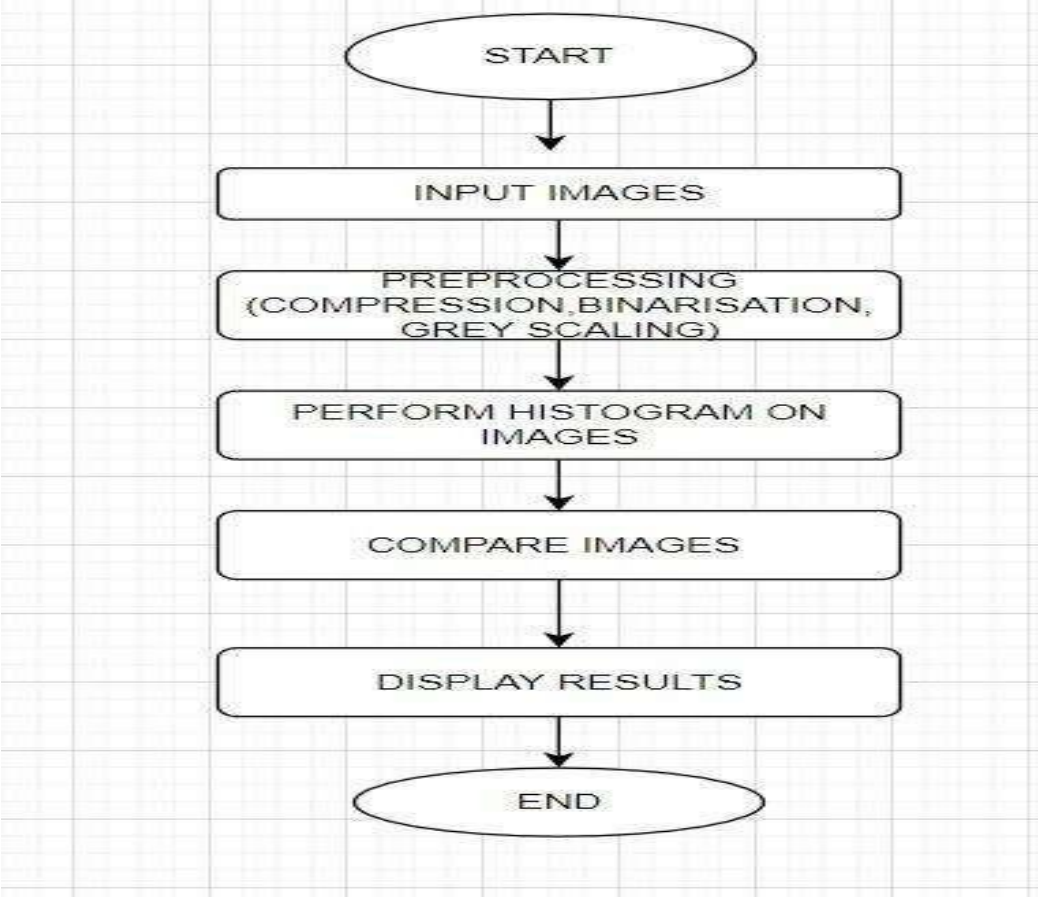
SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.ARCHITECTURE

3 ARCHITECTURE

3.1 SYSTEM ARCHITECTURE



3.1 system architecture

3.2 DESCRIPTION

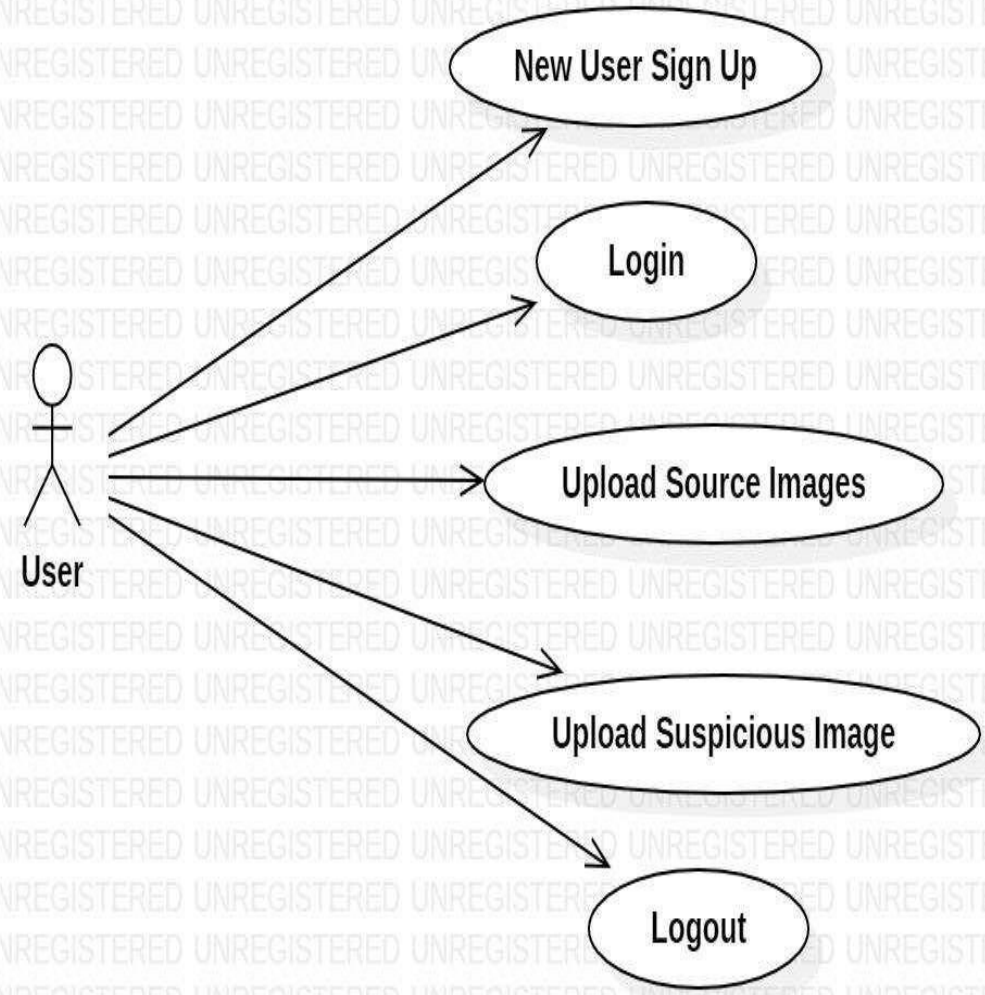
Input Data: Input data is given to the system to find the desired predicted output.

Data Preprocessing: It is the process in which the collected data is processed for the execution of the data.

Feature Selection: The features required for the prediction of image plagiarism.

3.2 USE CASE DIAGRAM

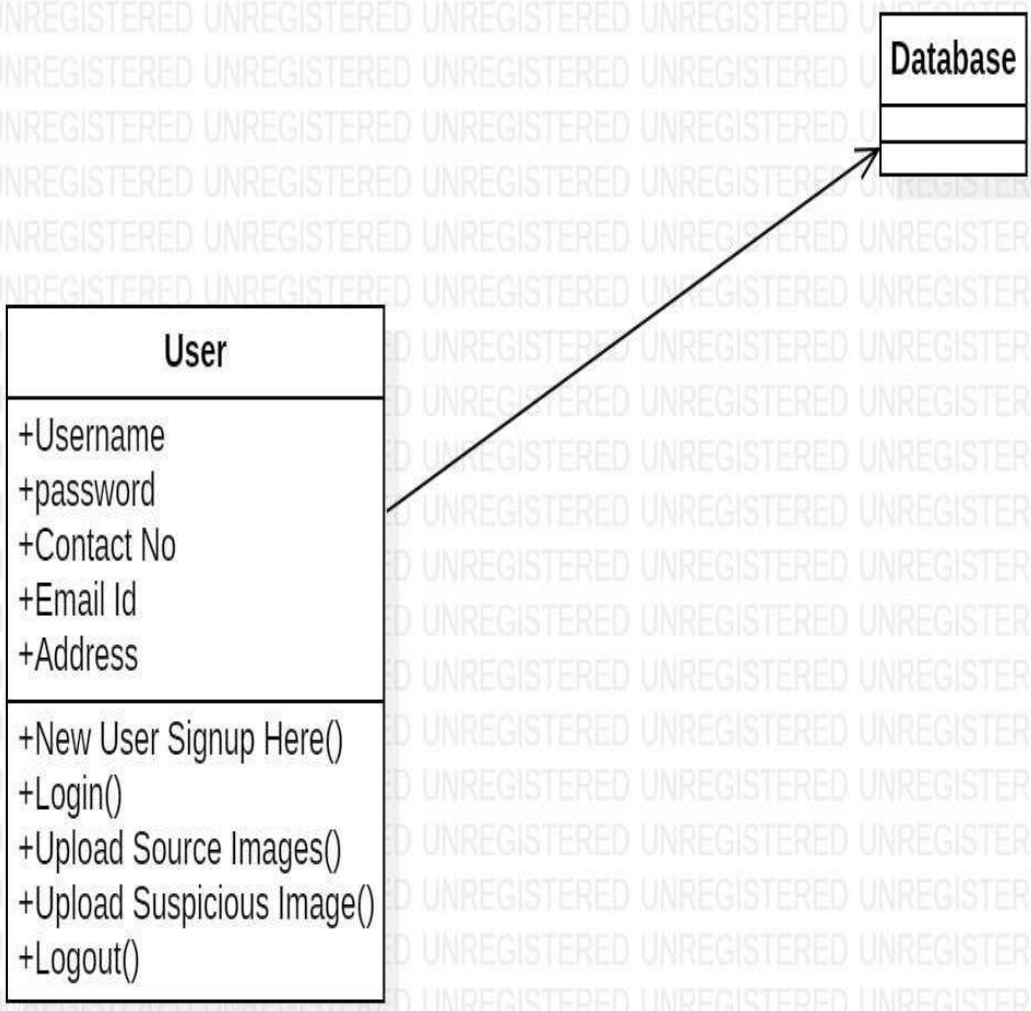
A use case diagram in the Unified Modeling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.



3.2 Use Case Diagram

3.3 CLASS DIAGRAM

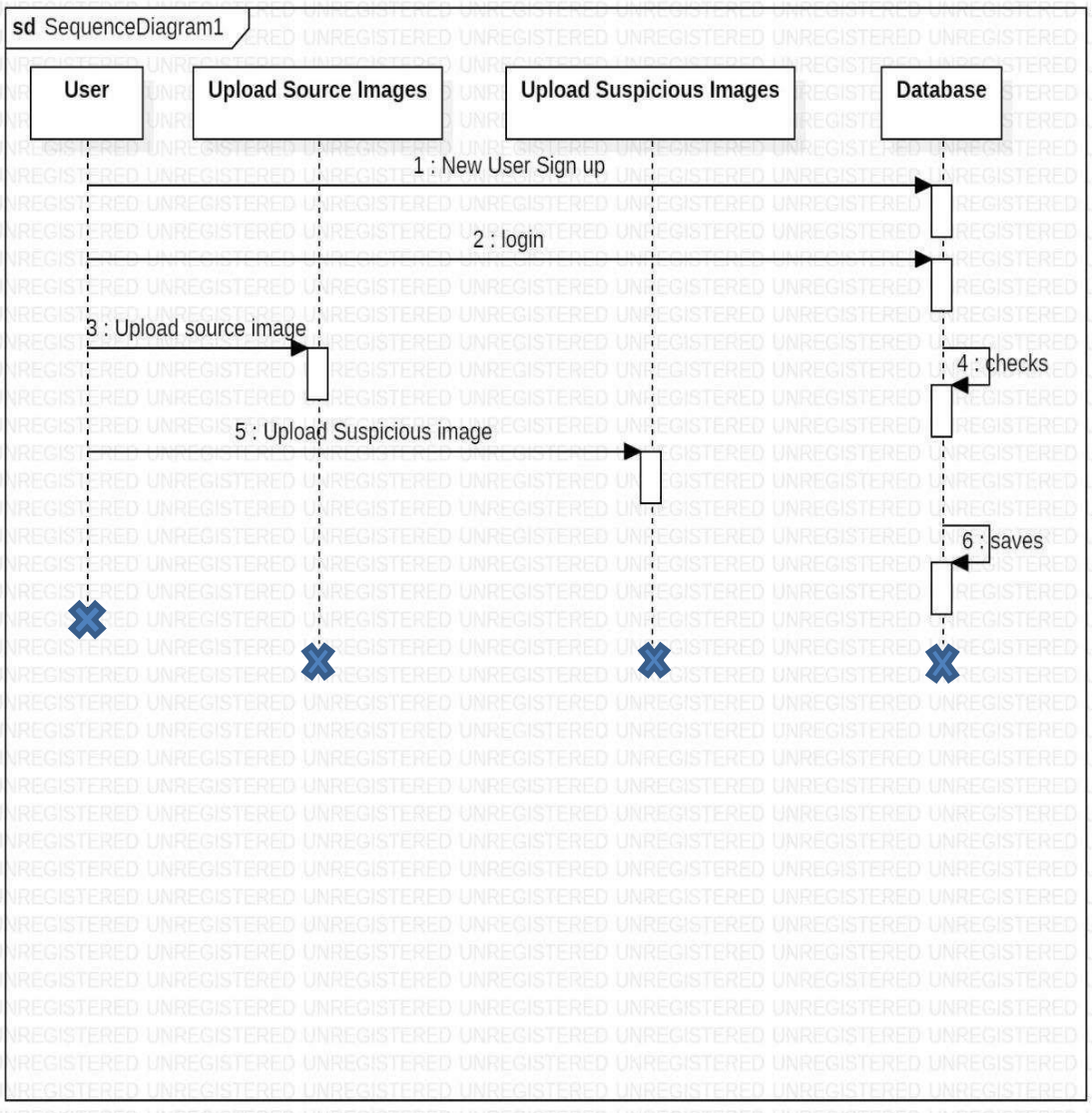
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



3.3 Class Diagram

3.4 SEQUENCE DIAGRAM

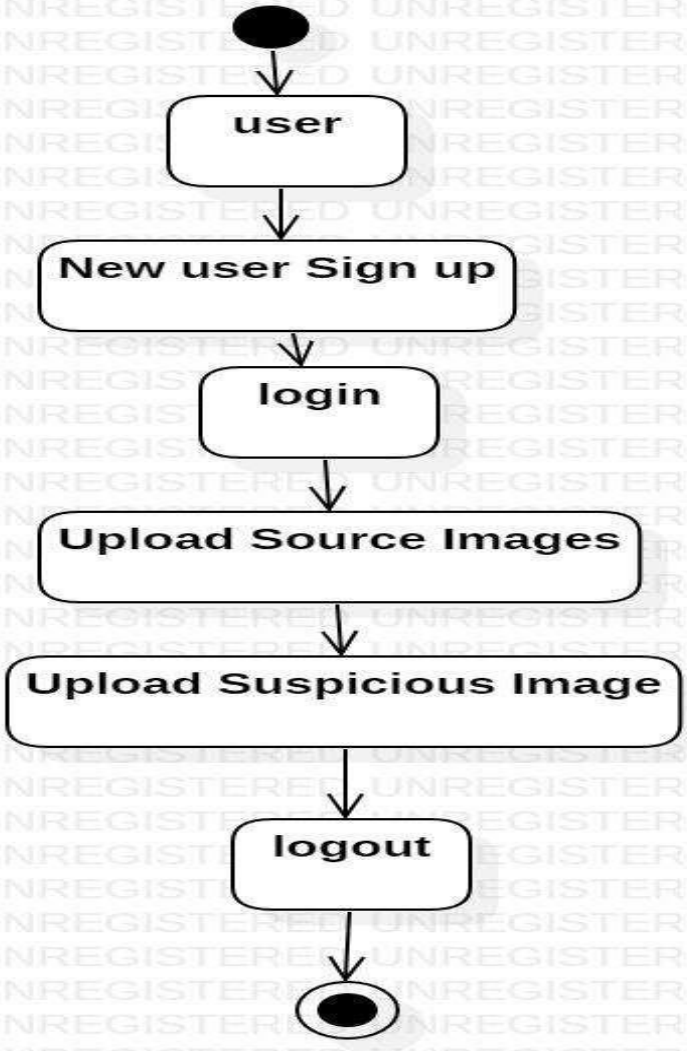
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



3.4 Sequence Diagram

3.5 ACTIVITYDIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.



3.5 Activity Diagram

4.IMPLEMENTATION

4.1 SAMPLE CODE

imported packages

```
from django.shortcuts import render
from django.template import RequestContext
from django.contrib import messages
import pymysql
from django.http import HttpResponse
from django.conf import settings
from django.core.files.storage import FileSystemStorage
import matplotlib.pyplot as plt
import re
import cv2
import numpy as np
from string import punctuation
from nltk.corpus import stopwords
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.stem import PorterStemmer
import os
from nltk.tokenize import word_tokenize

stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()
porter = PorterStemmer()
```

LCS method

```
def LCS(l1,l2):
    s1 = word_tokenize(l1)
```

```
    s2 = word_tokenize(l2)
    dp = [[None]*(len(s1)+1) for i in range(len(s2)+1)]
    for i in range(len(s2)+1):
        for j in range(len(s1)+1):
            if i == 0 or j == 0:
                dp[i][j] = 0
            elif s2[i-1] == s1[j-1]:
                dp[i][j] = dp[i-1][j-1]+1
            else:
                dp[i][j] = max(dp[i-1][j] , dp[i][j-1])
    return dp[len(s2)][len(s1)]
```

Tokenisation,stemming,lemmatisation,stop word removal

```
def cleanPost(doc):
    tokens = doc.split()
    table = str.maketrans('', '', punctuation)
    tokens = [w.translate(table) for w in tokens]
    tokens = [word for word in tokens if word.isalpha()]
    tokens = [w for w in tokens if not w in stop_words]
    tokens = [word for word in tokens if len(word) > 1]
    tokens = [lemmatizer.lemmatize(token) for token in tokens]
    tokens = [porter.stem(token) for token in tokens]
    tokens = ' '.join(tokens)
    return tokens
```

#Five modulus algorithm

```
def FMM(name):
    img = cv2.imread(name)
    rows,cols = img.shape
    for i in range(rows):
        for j in range(cols):
            if img[i,j] < 120:
                img[i,j] = 210
    for i in range(rows):
        for j in range(cols):
            k = img[i,j]
            if (k % 5) == 4:
                img[i,j] = k + 1
            elif (k % 5) == 3:
                img[i,j] = k + 2
            elif (k % 5) == 2:
                img[i,j] = k - 2
            elif (k % 5) == 1:
                img[i,j] = k - 1
    for i in range(rows):
        for j in range(cols):
            k = img[i,j]
            k = k / 5
            img[i,j] = k
    temp = img.ravel()
    temp = np.min(img)
    for i in range(rows):
```

```
        for j in range(cols):
            if img[i,j] > 0:
                img[i,j] = img[i,j] - temp
```

#plotting histogram

```
    hist = cv2.calcHist([img], [0], None, [256], [0, 256])
    return hist
```

#Suspicious Image processing

```
def UploadSuspiciousImageAction(request):
    if request.method == 'POST' and request.FILES['t1']:
        output = ''
        myfile = request.FILES['t1']
        fs = FileSystemStorage()
        name = str(myfile)
        filename = fs.save(name, myfile)
        hist = FMM(name)
        os.remove(name)
        similarity = 0
        file = 'No Match Found'
        hist1 = 0
        for i in range(len(image_files)):
            metric_val = cv2.compareHist(hist, image_data[i],
cv2.HISTCMP_INTERSECT)
            if metric_val > similarity:
                similarity = metric_val
                file = image_files[i]
                hist1 = image_data[i]
        output = '<table border=1 align=center><tr><th>Source Original Image
Name</th><th>Suspicious Image Name</th><th>Histogram Matching
Score</th><th>Plagiarism Result</th></tr>'
```

```
        result = 'No Plagiarism Detected'
        print(str(name)+" "+str(similarity))
        if similarity >= 39000:
            result = 'Plagiarism Detected'
        output+='<tr><td><font size="" color="white">'+file+'</td><td><font size="" color="white">'+name+'</td>'
        output+='<td><font size="" color="white">'+str(similarity)+'</td><td><font size="" color="white">'+result+'</td></tr>'
        context= {'data':output}
        fig, ax = plt.subplots(2,1)
        ax[0].plot(hist1, color = 'b')
        ax[1].plot(hist, color = 'g')
        plt.xlim([0, 256])
        ax[0].set_title('Original image')
        ax[1].set_title('Plagiarised image')
        plt.show()
        return render(request, 'SuspiciousImageResult.html', context)
```

#suspicious file accessing

```
def UploadSuspiciousFileAction(request):
    if request.method == 'POST' and request.FILES['t1']:
        output = ''
        myfile = request.FILES['t1']
        fs = FileSystemStorage()
        name = str(myfile)
        filename = fs.save("test.txt", myfile)
        data = ''
        with open("test.txt", "r", encoding='iso-8859-1') as file:
            for line in file:
                line = line.strip('\n')
                line = line.strip()
```

```
            data+=line+" "
        file.close()
        os.remove("test.txt")
        data = cleanPost(data.strip().lower())
        sim = 0
        ff = 'No Match Found'
        for i in range(len(text_data)):
            similarity = LCS(text_data[i],data)
            if similarity > sim:
                sim = similarity
                ff = text_files[i]


        output = '<table border=1 align=center><tr><th>Source Original File Name</th><th>Suspicious File Name</th><th>LCS Score</th><th>Plagiarism Result</th></tr>'
        result = 'No Plagiarism Detected'
        similarity_percent = 0
        if sim >= 0:
            similarity_percent = sim/len(word_tokenize(data))
            if similarity_percent >= 0.60:
                result = 'Plagiarism Detected'
        output+='<tr><td><font size="" color="white">'+ff+'</td><td><font size="" color="white">'+name+'</td>'
        output+='<td><font size="" color="white">'+str(similarity_percent)+'</td><td><font size="" color="white">'+result+'</td></tr>'
        context= {'data':output}
        return render(request, 'SuspiciousFileResult.html', context)



def UploadSourceImage(request):
    if request.method == 'GET':
```

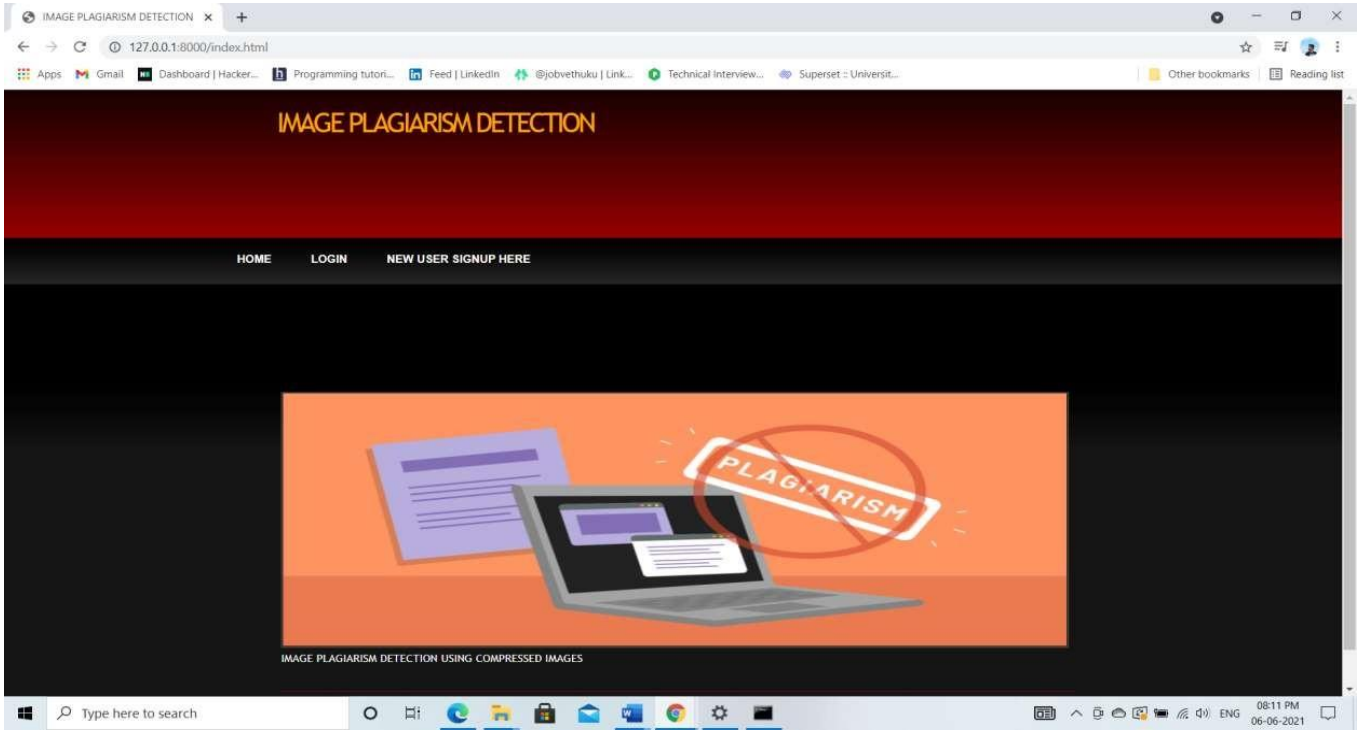
```
        if len(image_files) == 0:
            for root, dirs, directory in os.walk('images'):
                for j in range(len(directory)):
                    hist = FMM(root+"/"+directory[j])
                    image_data.append(hist)
                    image_files.append(directory[j])
        output = '<table border=1 align=center><tr><th>Source Image File Name</th><th>Histogram Values</th></tr>'
        for i in range(len(image_files)):
            output+='<tr><td><font size="" color="white">'+image_files[i]+'</td><td><font size="" color="white">'+str(image_data[i])+"</td></tr>"
        context= {'data':output}
        return render(request, 'UploadSourceImage.html', context)


def UploadSource(request):
    if request.method == 'GET':
        if len(text_files) == 0:
            for root, dirs, directory in os.walk('corpus-20090418'):
                for j in range(len(directory)):
                    data = ''
                    with open(root+"/"+directory[j], "r", encoding='iso-8859-1') as file:
                        for line in file:
                            line = line.strip('\n')
                            line = line.strip()
                            data+=line+" "
                    file.close()
                    data = cleanPost(data.strip().lower())
                    text_files.append(directory[j])
                    text_data.append(data)
        output = '<table border=1 align=center><tr><th>Source File Name</th><th>Words in File</th></tr>'
```

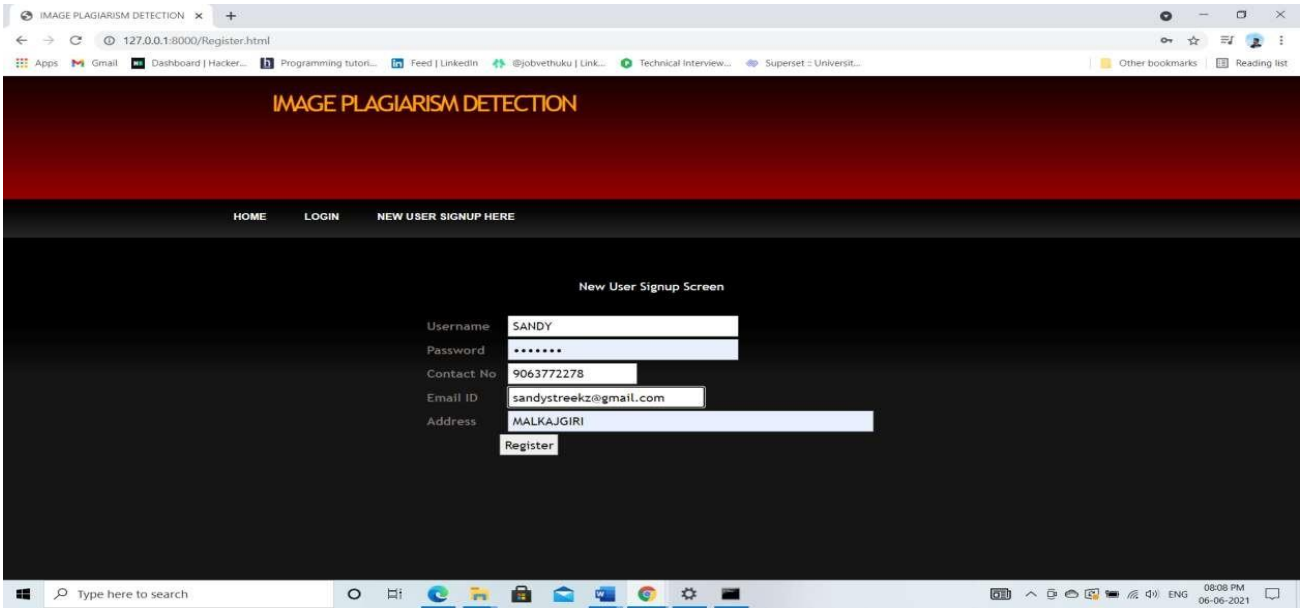
```
        for i in range(len(text_files)):
            length = len(text_data[i].split(" "))
            output+='<tr><td><font size="" color="white">'+text_files[i]+'</td><td><font size="" color="white">'+str(length)+"</td></tr>"
        context= {'data':output}
        return render(request, 'UploadSource.html', context)
```

5.SCREENSHOTS

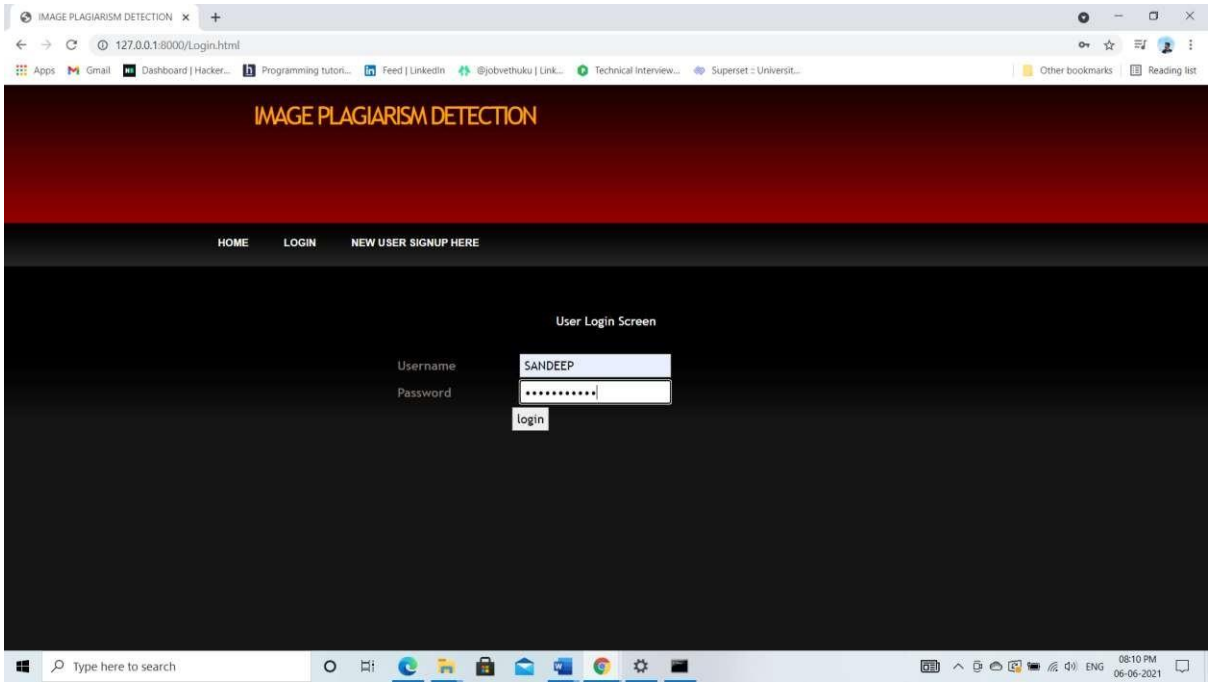
5.1 MAIN SCREEN AFTER STARTING THE PROGRAM



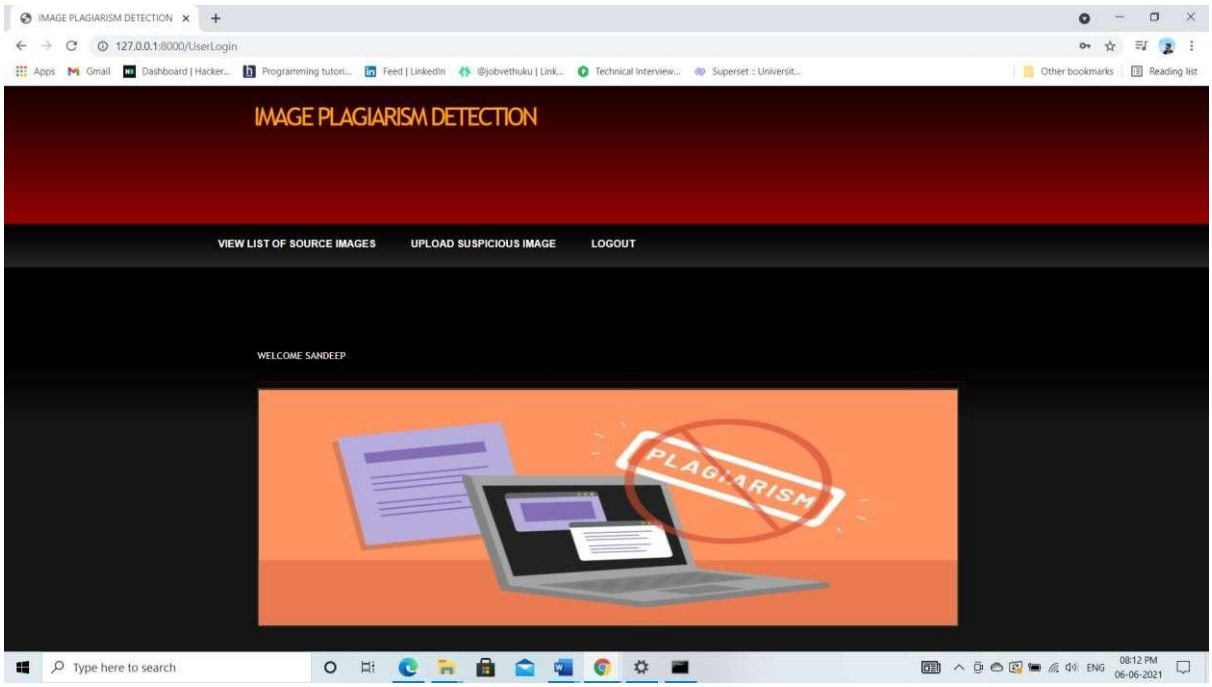
5.2 NEW USER SIGN UP



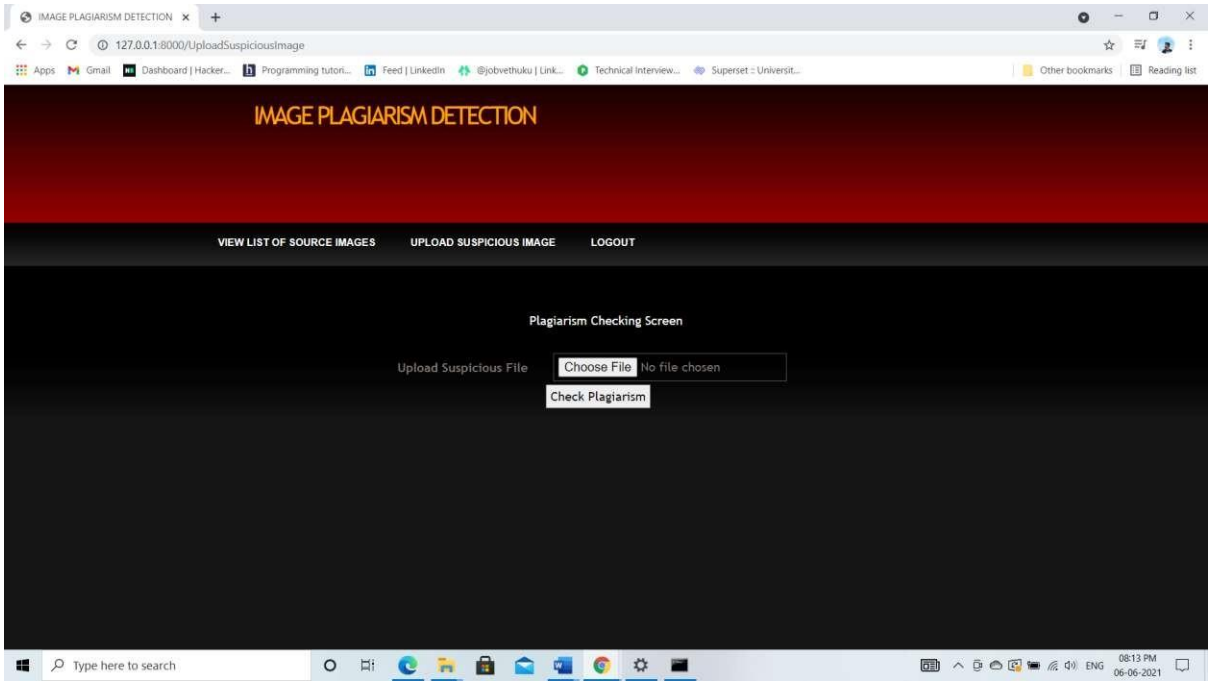
5.3 USER LOGIN



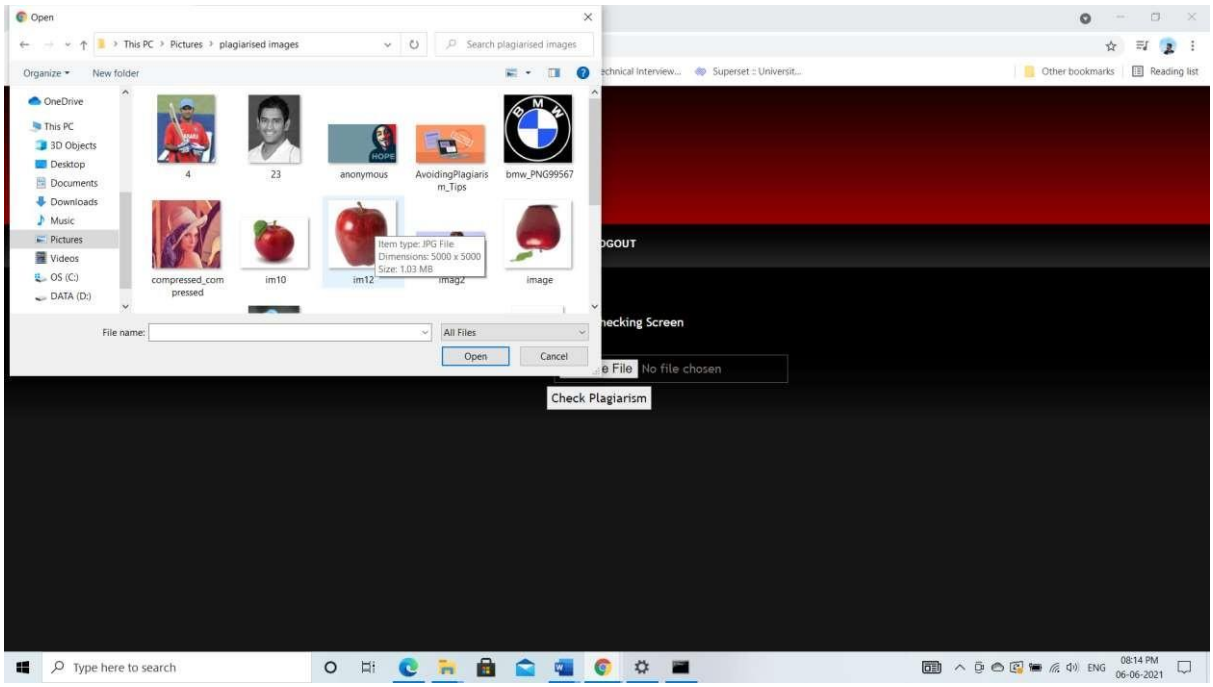
5.4 WELCOME SCREEN

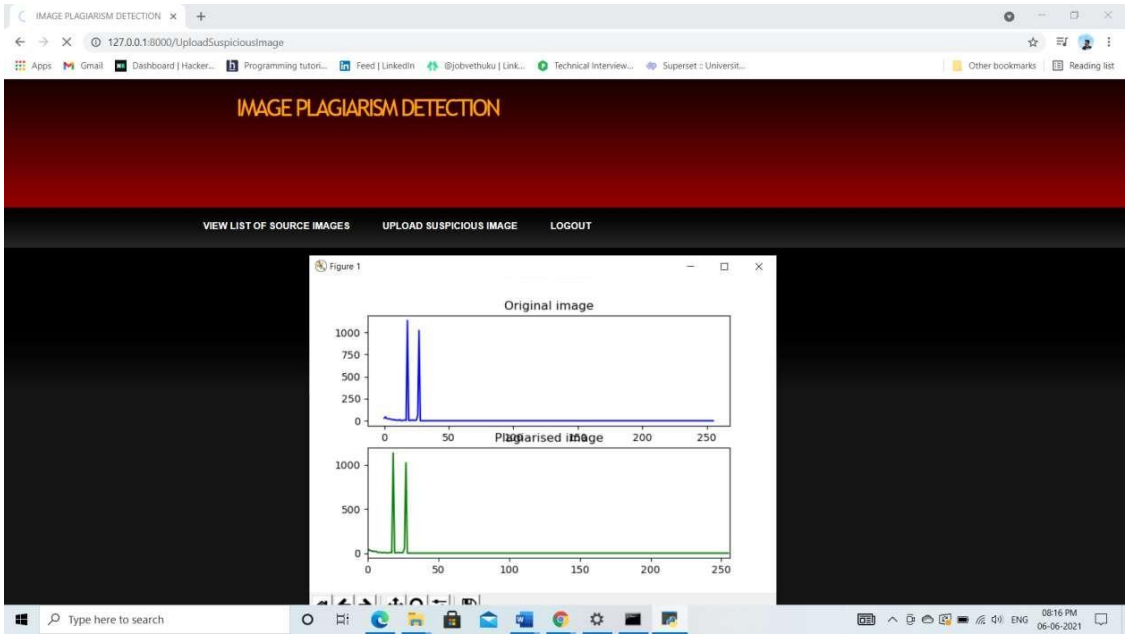


5.4 UPLOAD SUSPICIOUS IMAGE PAGE



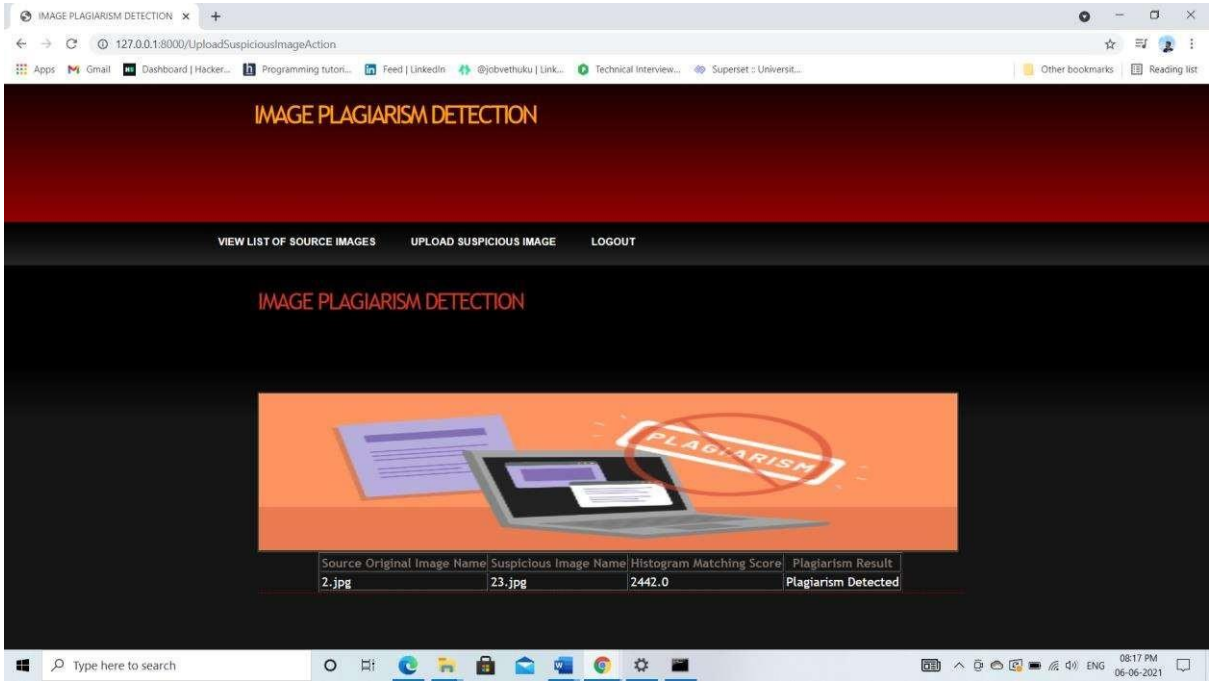
5.5 SELECT SUSPICIOUS IMAGE





5.6 HISTOGRAM OF SUSPICIOUS IMAGE AND SOURCE IMAGE

5.7 RESULT PAGE



6.TESTING

6.1 TESTING STRATEGIES

6.1.1 UNIT TESTING

Unit testing, a testing technique using which individual modules are tested to determine if there are issues by the developer himself.. it is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

Unit Testing Techniques:

Black Box Testing - Using which the user interface, input and output are tested.

White Box Testing –Used to test each one of those functions behavior is tested.

6.1.2 DATA FLOW TESTING

Data flow testing is a family of testing strategies based on selecting paths through the program's control flow in order to explore sequence of events related to the status of Variables or data object. Dataflow Testing focuses on the points at which variables receive and the points at which these values are used.

6.1.3 INTEGRATION TESTING

Integration Testing done upon completion of unit testing, the units or modules are to be integrated which gives raise too integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

6.1.4 BIG BANG INTEGRATION TESTING

Big Bang Integration Testing is an integration testing Strategy wherein all units are linked at once, resulting in a complete system. When this type of testing strategy is adopted, it is difficult to isolate any errors found,

because attention is not paid to verifying the interfaces across individual units.

6.1.5 USER INTERFACE TESTING

User interface testing, a testing technique used to identify the presence of defects is a product/software under test by Graphical User interface [GUI].

6.2 TEST CASES

● REGISTRATION TEST CASE

Test Case	Input	Test case Description	Expected Output	Actual Output	Status
1	Invalid user id and password	User registration	Unable to Register	Displays message to choose different username	Pass
2	Valid User id and password	User registration	Registration Successful	user registered successfully	Pass

Table 4.1: Test case for user registration

● LOGIN TEST CASE

Test Case	Input	Test case Description	Expected Output	Actual Output	Status
1	Invalid user id and password	User registration	Unable to Register	Displays message to choose different username	Pass
2	Valid User id and password	User registration	Registration Successful	user registered successfully	Pass

Table 4.2 : Test case for user login

7.CONCLUSION

5.1 CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

It has been a pleasure for me to work on this exciting and challenging project.

This project proved good for me as it provided practical knowledge of not only

Programming in python & MYSQL.It has provides knowledge about the latest technology used in developing web enabled applications and client server technology that will be in great demand in future.This will provide better opportunities and guidance in future in developing projects independently.Tools used to develop this project are Django,python 3.7,MYSQL,SQLYog,HTML and CSS.Image plagiarism can be detected even when the image color is changed or even if the image is rotated.

There werw no errors observed.

5.2 FUTURE SCOPE

Image plagiarism accuracy is about 80 percent which can be improved further by implementing it with deep learning techniques and extracting images from documents can be done in future scope.

8.BIBILIOGRAPHY

REFERENECES

1.B.Hadi and M.J.Kargar.”Plagiarism detection of flow chart images in the texts”,2017 3th International Conference on Web Research(ICWR),2017,pp,128-132

2.P.Ovhal, “Detecting plagiarism in images,”,2015 International conference on Information Processing (ICIP),2015.

3.Firas A.Jassim and Hind E.Qasim,”Five modulus method for Images compression”, An Internatinal journal (SIPIJ), 2012.

4. Wang Wen, Wang yanb and Li Bingbing , “Research on Plagiarism IdentificationofDigtialImage” IEEE,2010.

5.Harpreet Kaur and Neelofar Sohi ,”A Study Application of Histogram in Image Enhancement,” he International journal of Engineering and Science (IJES), 2017 PP:59-63

6 Jithin S Kurvilla, Midhun Lal V L, Rejin Ro, Tomin Baby, Sangeetha jamal and Shely kk,” Flow chart Plagiarism Detection System : An Image Processing Approach” , 7th International Conference on Advance in Computing Communication,2017

8.2 GITHUB LINK

https://github.com/johnnova17/IMAGE-PLAGIARISM