CS311 Analysis Of Algorithm

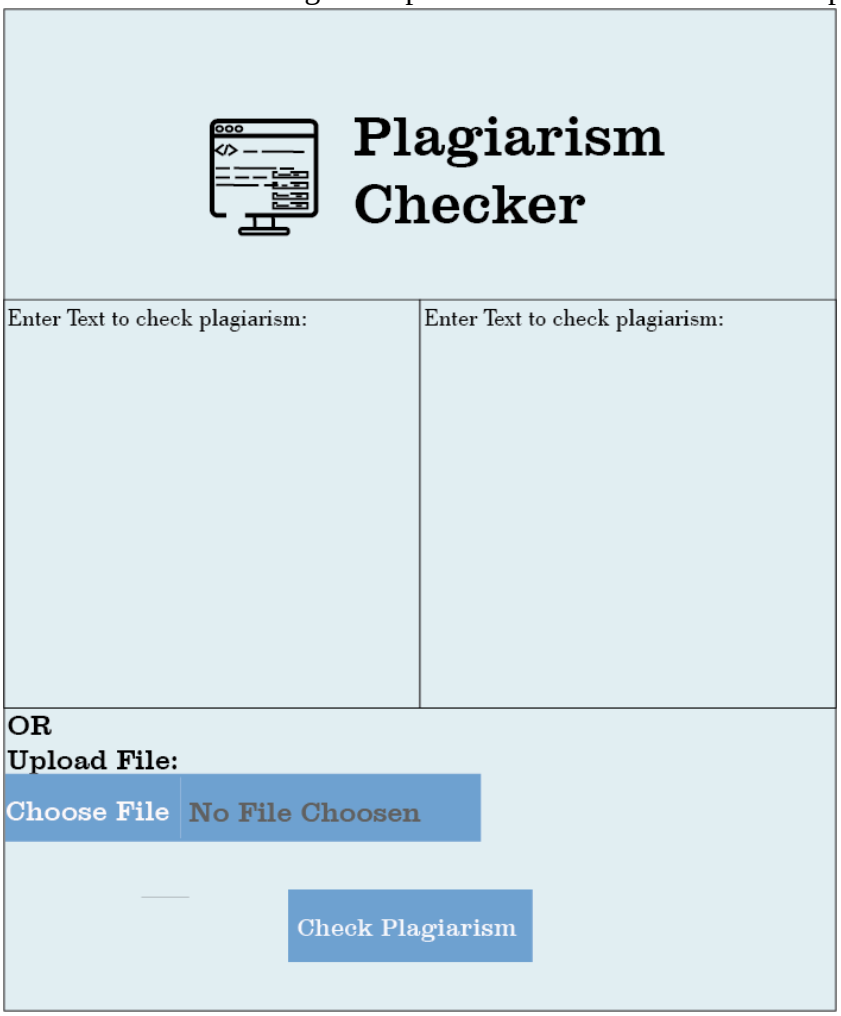
Term Project

2018-CS-124

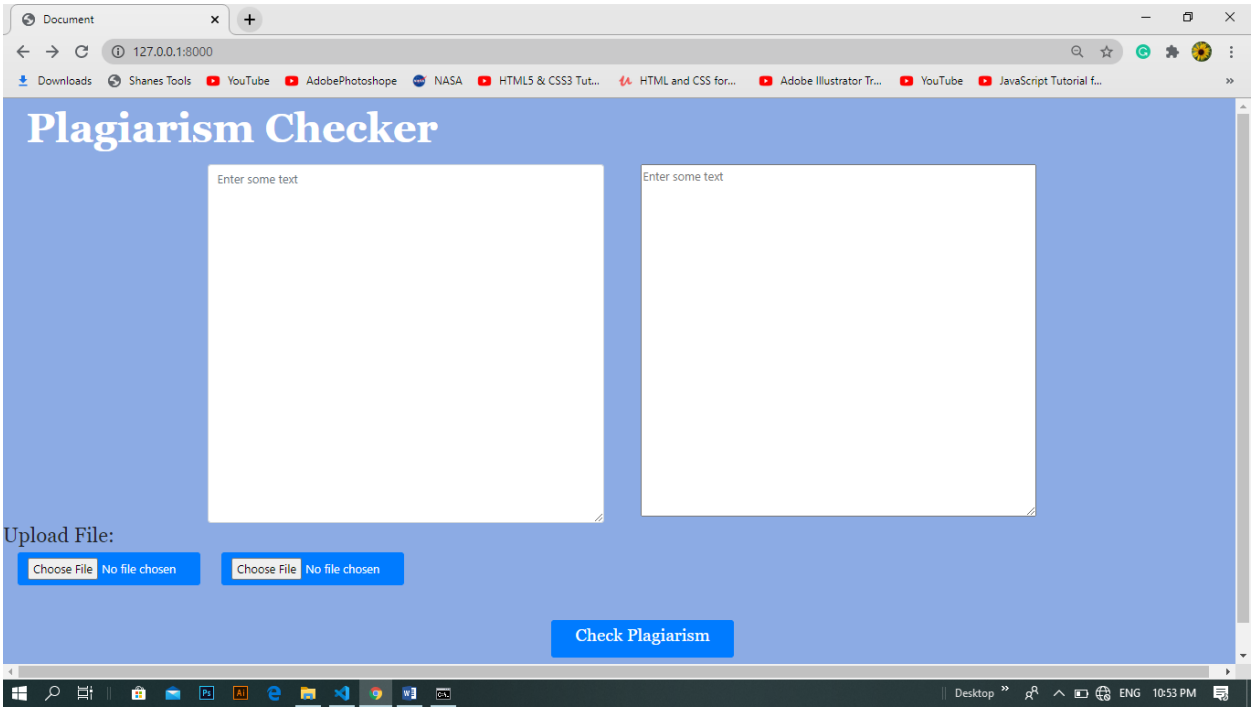
2018-CS-131

This document contains all the information about interfaces design and the integration of my code file with the interface.

We are using django for building the interface and for the integration of python code file with our interface. Following is the picture of our interface we are supposed to deign.



Since we use django our design interface looks bit different from this one. Following is the interface design by us using django:



Here user will enter text in both these text areas. Folllowing is the html file code named as 'home.html' in our project folder.

home.html file:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-
```

Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

```
<style>
  .inputfile{
    width: 0.1px;
    height: 0.1px;
    opacity: 0;
    overflow: hidden;
    position: absolute;
    z-index: -1;
  }
  .inputfile + label{
    font-size:1.25em;
    font-weight: 700;
    color:white;
    background-color: blue;
    display: inline-block;
  }
  .inputfile:focus +label,
    .inputfile+label:hover{
      background-color: rgb(71, 97, 214);


  }
</style>
<title>Document</title>
</head>
<body style="background-color: rgb(140, 171, 228);">
    <nav class="navbar navbar-expand-lg">
```

```
<a class="navbar-
brand" href="#"><img src="C:\Users\ayesha\Desktop\vac\graphic course19\PlagiarismChecker l
ogo.png" alt=""></a>

<button class="navbar-toggler" type="button" data-toggle="collapse" d

ata-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">

  <span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

  <ul class="navbar-nav">

    <li class="nav-item">

      <a class="nav-item display-4 text-sm-left font-weight-bold text-md-left" style="font-
family:Georgia, 'Times New Roman', Times, serif;color: black; text-
decoration: none;"  href="#">Plagiarism Checker</a>

    </li>

  </ul>

</div>

</nav>

<h1 class="text-center"></h1>

<div class="conatainer-fluid">

<form action="check">

  <div class="row arow">

    <div class="col-sm-2 col-md-4"><textarea class="form-control" name="user-
msg1" id="" cols="100" rows="20" placeholder="Enter some text"  style="width: 100%;margin-
left:280px;"></textarea></div>

    <div class="col-sm-2 col-md-4"><textarea name="user-
msg2" id="" cols="140" rows="20" placeholder="Enter some text" style="width: 100%;margin-
left:300px;"></textarea></div>


  </div>
```

```
        </div>

        <h3 style="font-family: Georgia, 'Times New Roman', Times, serif;">Upload File:</h3>

        <input class="btn btn-primary" style="width: 250px; margin-
left: 20px;"   type="file" name="file1" class="inputtfile"/>

        <label for="file"></label>

        <input class="btn btn-primary" style="width: 250px; margin-
left: 20px;"   type="file" name="file2" class="inputtfile"/>

        <br><br><br>

        <button type="submit" class="btn btn-primary col-sm-3" style="margin-
left:750px;width: 250px;"><h4 style="font-
family: Georgia, 'Times New Roman', Times, serif;">Check Plagiarism</h4></button>

       </form>

<br><br><br>

         <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integr
ity="sha384-
wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigi
n="anonymous"></script>

    </body>

    </html>
```

After this **views.py** file is the python file having the code to find the plagiarized text. Here is the code for

View.py file

```
from django.shortcuts import render

from django.http import HttpResponse

# Create your views here.


MAX_SIZE = 10000

res=0

resultArray = [" "for i in range(MAX_SIZE)]

def home(request):
```

```
    global res
    res=0
    return render(request,'home.html')
def rename(request):
    file1=request.GET[file1]
    file2=request.GET[file2]
    k=len(file1)
    return render(request,'f2f.html',{'result':k})
my=12
#resultArray=["for i in range(MAX_SIZE)]
def LongestommonSubstring(str1,str2, n1, n2):            #it is a function  that return the number of matched characters
    global memoizedArray
    global resultArray
    global k
    if (n1 == 0 or n2 == 0):
        return 0
    if (memoizedArray[n1 - 1][n2 - 1] != -1):       #if computed already then simply return that computed value
        return memoizedArray[n1 - 1][n2 - 1]
    if (str1[n1 - 1] == str2[n2 - 1]):   #case1: if matches then add 1 and call it self with length - 1
        #resultArray[k]=str1[n1-1]
        #k=k+1
        memoizedArray[n1 - 1][n2 - 1] = 1 + LongestommonSubstring(str1, str2, n1 - 1, n2 - 1)
        return memoizedArray[n1 - 1][n2 - 1]
    else:                                   #case2: if dunno match
        memoizedArray[n1 - 1][n2 - 1] = max(LongestommonSubstring(str1, str2, n1, n2 - 1),
                                LongestommonSubstring(str1, str2, n1 - 1, n2))
```

```
        return memoizedArray[n1 - 1][n2 - 1]
def dataReadingAndLCS(file1, file2):
    global resultArray
    global memoizedArray
    global res
    char_arr1 = []
    char_arr2 = []
    string1 = []
    string2 = []
    f1 = file1
    f2 = file2
    l1 = len(f1)
    l2 = len(f2)
    for i in range(0, l1):
        temp=0
        if f1[i] != '.':
            char_arr1.append(f1[i])
        else:
            string1 = ""
            for x in char_arr1:
                string1 += x
            for f in range(0, len(char_arr1)):
               if (char_arr1[f] == ' '):
                  char_arr1[f] = '\0'
            for j in range(0, l2):

                if f2[j] != '.':
                    char_arr2.append(f2[j])
```

```
            else:
                string2 = ""
                for y in char_arr2:
                    string2 += y
                #temp = 0
                #for f in range(0, len(char_arr1)):
                 #   if (char_arr1[f] == ' '):
                  #      char_arr1[f] = '\0'

                for f in range(0, len(char_arr2)):
                    if (char_arr2[f] == ' '):
                        char_arr2[f] = '\0'
                s=LongestommonSubstring(char_arr1,char_arr2,len(char_arr1),len(char_arr2))
                if(s>temp):
                    temp =s
                    char_arr2 = []
                    char_arr1 = []
            resultArray = string2
        #print(string2, end=" ")
        res += temp
        my=res
        print("sdf")
        print(my)
def check(request):
    MAX_SIZE=1000
    global memoizedArray
    global res
    global resultArray
```

```
    k=0
    #resultArray = [" "for i in range(MAX_SIZE)]
    val1=request.GET['user-msg1']
    val2=request.GET['user-msg2']
    memoizedArray = [[-1 for i in range(MAX_SIZE)]
            for i in range(len(val1))]
    dataReadingAndLCS(val1,val2)
    print("matched words are: ")
    print(resultArray)
    print("\n Length is ")
    print(res)
    if (len(val1)==0):
        val1=1

    percent=res/len(val1)*100

    return render(request,'Results.html',{'result':res,'resultarray':resultArray,'percent':percent})
```

urls.py file contains urls for all the functions we make in views.py.

urls.py

```
from django .urls import path
from . import views
urlpatterns=[
path("",views.home,name='home'),
path("check",views.check,name='check'),
path("rename",views.rename,name='rename')
]
```

manage.py is the main file which we use to run the server by writing ‘’python manage.py runserver’’ on command prompt .

manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'plagiarismcheckingtool.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
```

models.py:

```
from django.db import models

# Create your models here.
```

apps.py:

```
from django.apps import AppConfig


class CalcConfig(AppConfig):
    name = 'calc'
```

admin.py

```
from django.contrib import admin


# Register your models here.
```

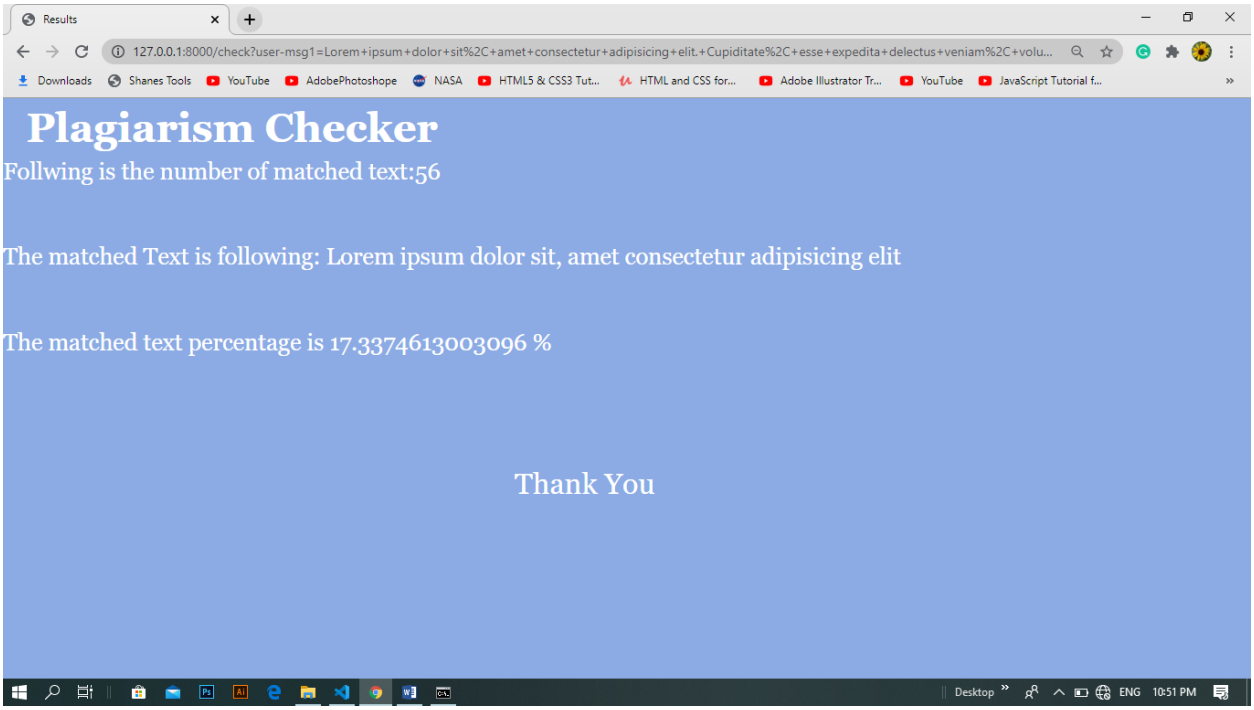
Results.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.
min.css" integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossori
gin="anonymous">
    <title>
        Results
    </title>
</head>
<body style="background-color: rgb(140, 171, 228);color:white;">
    <nav class="navbar navbar-expand-lg">
```

```
      <a class="navbar-
brand" href="#"><img src="C:\Users\ayesha\Desktop\vac\graphic course19\PlagiarismChecker l
ogo.png" alt=""></a>

      <button class="navbar-toggler" type="button" data-toggle="collapse" d

      ata-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">

        <span class="navbar-toggler-icon"></span>

      </button>

      <div class="collapse navbar-collapse" id="navbarNav">

        <ul class="navbar-nav">

          <li class="nav-item">

            <a class="nav-item display-4 text-sm-left font-weight-bold text-md-left" style="font-
family:Georgia, 'Times New Roman', Times, serif;color:white; text-
decoration: none;"  href="#">Plagiarism Checker</a>

          </li>

        </ul>

      </div>

     </nav>

      <h2 style="color:white;font-
family: Georgia, 'Times New Roman', Times, serif;">Follwing is the number of matched text:{{r
esult}}</h2><br><br><br>

      <h2 style="color:white;font-
family:Georgia, 'Times New Roman', Times, serif">The matched Text is following: {{resultarra
y}}</h2>

      <br><br><br>

      <h2 style="color: white;font-
family: Georgia, 'Times New Roman', Times, serif;">The matched text percentage is {{percent}
} %</h2>

      <br><br><br><br><br><br>

      <h1 style="margin-left: 700px;color:white;font-
family: Georgia, 'Times New Roman', Times, serif;">Thank You</h1>

</body>
```

</html>

this file then give us a interface with results render by view.py



f2f.html:

this file show results when files are selected from browse button.

Result:{{result}}

wsgi.py:

"""

WSGI config for plagiarismcheckingtool project.

It exposes the WSGI callable as a module-level variable named ``application``.

```
For more information on this file, see
https://docs.djangoproject.com/en/3.0/howto/deployment/wsgi/
"""

import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'plagiarismcheckingtool.settings')

application = get_wsgi_application()
```

settings.py:

```
"""
Django settings for plagiarismcheckingtool project.

Generated by 'django-admin startproject' using Django 3.0.8.

For more information on this file, see
https://docs.djangoproject.com/en/3.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.0/ref/settings/
"""

import os
```

```
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '0i&@e^_)mlq@f2!siybz6^6!cckuxn+akqf)ax@r-#wu%qo55&'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'plagiarismcheckingtool.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR,'Templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

```
WSGI_APPLICATION = 'plagiarismcheckingtool.wsgi.application'



# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}



# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
```

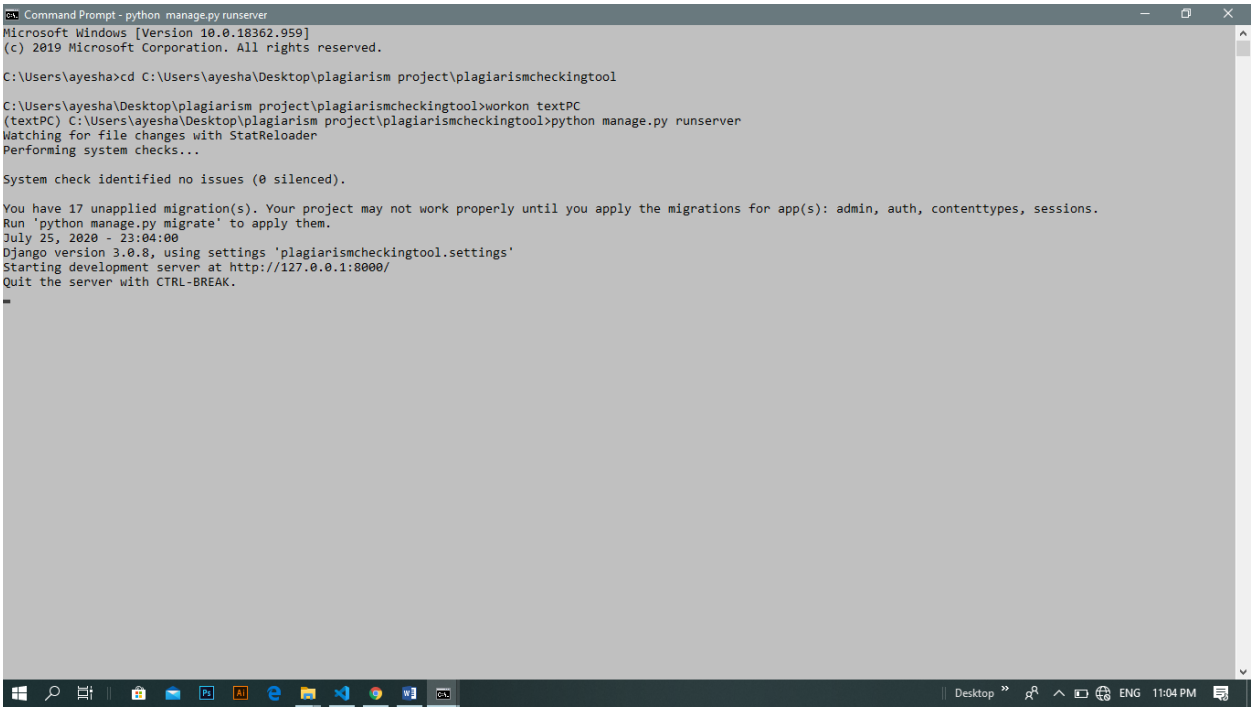
```
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]



# Internationalization
# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True



# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.0/howto/static-files/

STATIC_URL = '/static/'
```

asgi.py file:

```
"""
ASGI config for plagiarismcheckingtool project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/3.0/howto/deployment/asgi/
"""

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'plagiarismcheckingtool.settings')

application = get_asgi_application()
```

Our app project folder have four more files in “ __pycache__” .

We have make a virtual environment named as “textPC” in our local machine to run this project. Following picture is the screenshot of command prompt having commands use to run the server.



Implementation code:

```
MAX_SIZE = 10000
resultArray=['''for i in range(MAX_SIZE)]
def LongestommonSubstring(str1,str2, n1, n2):          #it is a function  that return the number of matched characters
    global memoizedArray
    global resultArray
    global k
    if (n1 == 0 or n2 == 0):
        return 0
    if (memoizedArray[n1 - 1][n2 - 1] != -1):      #if computed already then simply return that computed value
        return memoizedArray[n1 - 1][n2 - 1]
    if (str1[n1 - 1] == str2[n2 - 1]):   #case1: if matches then add 1 and call it self with length - 1
        #resultArray[k]=str1[n1-1]
        #k=k+1
        memoizedArray[n1 - 1][n2 - 1] = 1 + LongestommonSubstring(str1, str2, n1 - 1, n2 - 1)
        return memoizedArray[n1 - 1][n2 - 1]
    else:                                  #case2: if dunno match
        memoizedArray[n1 - 1][n2 - 1] = max(LongestommonSubstring(str1, str2, n1, n2 - 1),
                                            LongestommonSubstring(str1, str2, n1 - 1, n2))

        return memoizedArray[n1 - 1][n2 - 1]
```

```
def dataReadingAndLCS(file1, file2):                    # function to check the match text
    global resultArray
    global f1, f2
    global res
    char_arr1 = []
    char_arr2 = []
    string1 = []
    string2 = []
    f1 = file1.read()
    f2 = file2.read()
    l1 = len(f1)
    l2 = len(f2)
    for i in range(0, l1):
        temp=0
        if f1[i] != '.':
            char_arr1.append(f1[i])
        else:
            string1 = ""
            for x in char_arr1:
                string1 += x
            for f in range(0, len(char_arr1)):
                if (char_arr1[f] == ' '):
                    char_arr1[f] = '\0'
            for j in range(0, l2):

                if f2[j] != '.':
                    char_arr2.append(f2[j])
                else:
                    string2 = ""
                    for y in char_arr2:
                        string2 += y
                    #temp = 0
                    #for f in range(0, len(char_arr1)):
                     #   if (char_arr1[f] == ' '):
                      #      char_arr1[f] = '\0'

                    for f in range(0, len(char_arr2)):
                        if (char_arr2[f] == ' '):
                            char_arr2[f] = '\0'
                    s=LongestommonSubstring(char_arr1,char_arr2,len(char_arr1),len(char_arr2))
                    if(s>temp):
                        temp =s
                        char_arr2 = []
```

```
                char_arr1 = []
        resultArray = string2
        #print(string2, end=" ")
        res += temp
def CPPPlagiarismmChecking(file1,file2):      #  function to check plagiarism between files
    global resp
    global f1, f2
    char_arr1 = []
    char_arr2 = []
    f1 = file1.read()
    f2 = file2.read()
    l1 = len(f1)
    l2 = len(f2)
    for i in range(0, l1):

        if f1[i] != '.':
            char_arr1.append(f1[i])
        else:
            string1 = ""
            for x in char_arr1:
                string1 += x
            for j in range(0, l2):

                if f2[j] != '.':
                    char_arr2.append(f2[j])
                else:
                    string2 = ""
                    for y in char_arr2:
                        string2 += y
                    LongestommonSubstring(string1,string2,len(string1),len(string2))
                    char_arr2 = []
            char_arr1 = []


res = 0
k=0
doc1 = open('text1.txt', 'r')
textt1 = doc1.read()
doc1.close()
resultArray = [" "for i in range(MAX_SIZE)]
doc2 = open('text2.txt', 'r')
memoizedArray = [[-1 for i in range(MAX_SIZE)]
      for i in range(len(textt1))]
```

```
doc1=open('text1.txt','r')
dataReadingAndLCS(doc1, doc2)
print("matched words are: ")
print(resultArray)
# #for i in range(len(resultArray)):
#  #   if(resultArray[i]!=' '):
#   #      print(resultArray[i], end=" ")
print("\n Length is ")
print(res)
doc1.close()
doc2.close()
print(res/len(textt1)*100)
```

Thank you.