

Apeejay College of Fine Arts, Jalandhar

PROJECT REPORT

ON

Web based Application to detect Plagiarism: PlagiDetect



Submitted

To

Guru Nanak Dev University, Amritsar

For the partial fulfilment of the degree of M.Sc. (IT)-II

Supervised By: Submitted By:

Mr. Jagmohan Mago Simran Uppal

(Roll no. 20861924556)

Simranjeet Kaur

(Roll no. 20861924554)

Apeejay College of Fine Arts, Jalandhar



Department of Computer Science

This is to certify that the project entitled "Web based Application to detect Plagiarism: Plagidetect", submitted by Simran Uppal (Roll no. 20861924556) and Simranjeet Kaur (Roll no. 20861924554) in the partial fulfilment of the award of the degree of M.Sc. IT-II is an authentic piece of work carried out under our supervision and guidance.

The matter embodied herein has not been submitted to any other university for the award of any other degree.

Dr. Neerja Dhingra Dr. Roopali Sood

(Principal) (H.O.D)

Mr. Jagmohan Mago

(Project Guide)

ACKNOWLEDGEMENT

We are profoundly grateful to all the wonderful people who supported and guided us during the course of our project work. They not only encouraged us to keep moving forward towards the project completion, but also taught us valuable lessons of life such as persistence, hard work, determination, and positive approach. The knowledge and support that we have received shall remain with us for the whole life and help us in facing real life problems in any field we work in the future. Their seriousness and inspiration helped us a lot in achieving our project goal with passion. We are short of words to express our heartiest gratitude to all of them.

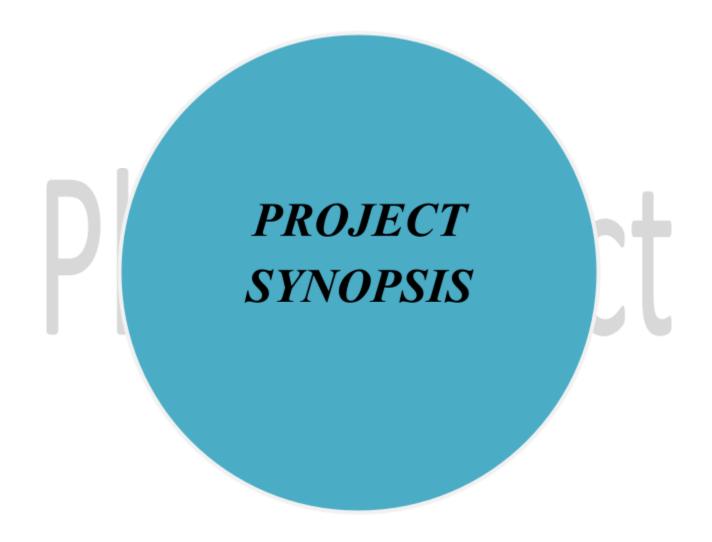
We take this opportunity to express a deep sense of gratitude to Dr. Neerja Dhingra (Principal), Dr. Roopali Sood (HOD Computer Science Department), Mr. Jagmohan Mago (Project Guide), and the whole computer staff for assisting in the fulfilment of our project, and giving us their valuable time and knowledge despite their busy schedules. We thank our project guide in encouraging us to choose a project topic that would build us professionally as well as personally. Keeping all the aspects in mind, they guided us from time to time, and also provided additional support from the teachers of computer staff who assisted us in particular problems of our project. We are obliged to all those staff members (Ms. Poonam, Ms. Pratibha, and Mr. Yogesh) for providing exact solutions to our specific project problems. We thank everyone for giving their appropriate advice to us.

Simran Uppal

Simranjeet Kaur

	INDEX	
Serial Number	Торіс	Page Number
1	Project Synopsis	1-3
2	Project Definition and Objective	4-5
3	System Study and Analysis	6-8
4	Proposed System	9-11
5	Design Tools	12-15
6	Working Methodology	16-22
7	Coding and Layouts	23-48
8	Testing	49-51
9	Security Mechanism	52-54
10	Implementation and Maintenance	55-56
11	Contribution	57-58
12	Conclusion	59-60

PlagiDetect



PROJECT TITLE: PlagiDetect

PROJECT DEFINITION

The project is a web application of plagiarism detection. Plagiarism refers to the pilfering of another person's writing into one's piece of work, which can be an essay, assignment, or research paper without attributing the original writer. The primary expectation in every class is that whatever you write will be your own words, generated from your understanding. Plagiarists perform this unacceptable activity and pass the respective stage without being caught. It becomes very tiring for the professors or checkers to identify the possible cribbing involved. PlagiDetect is a solution to the problem of plagiarism. It will provide the facility of detecting plagiarism in the text with flexible options of accessing text files.

OBJECTIVE

The main objective of this project is to create a web application to detect plagiarism. The registered user should receive a detailed report of the result that PlagiDetect will generate. The end users will have to register to PlagiDetect by creating an account. Users will be able to give access to the text files with four options: upload the text files from any location on the computer into PlagiDetect web application, enter a particular drive name in which all the available text files will be processed, enter a particular path to a folder name, and to check whole of the local computer for the text files. In this way, a professor can check the students' assignments in a fraction of a second. Plagiarism detection is a great way to check the originality of a document.

The results of each plagiarism detection should be read and understood to address the shortcomings in the composition.

WORKING METHODOLOGY

Agile methodology is the ideal approach for the PlagiDetect application. It focuses on building a single feature at a time. Agile uses sprints or a set of times when a particular trait of the application should be developed, tested, and presented. It incorporates the entire Software Development Life Cycle (SDLC) for a feature into each sprint. It helps to stick to a planned schedule but also allows for repeated analyses. The broad idea is to split the entire project into multiple features (four in this case) through which the same set of steps follow. This approach is known as iterative and incremental development. The increment happens from one component to another, and repeated execution is performed iteratively for each part or module. These modules further comprise sub-modules. At every level, a similar approach follows. Python is used at the backend of a Django web application to generate the logic for accessing files in different ways and ultimately check similarity among all the text files. The source-code editor used for this web application is Visual Studio Code.

CONTRIBUTION

A web application like PlagiDetect is need of the hour. It will contribute to the professional organizations like educational institutes in checking assignments. Also, it will assist researchers in getting the plagiarism percentage in their work. It will help eradicate academic dishonesty and contract cheating. The possibility of plagiarism can arise whether a single person performs it or multiple students contact the same person to complete their assignments. Whether knowingly or

unknowingly, a piece of text can be prone to plagiarism. It is always significant as well as safe to apply plagiarism detection on a particular piece of writing.



PROJECT DEFINITION AND OBJECTIVE

The project is a web application of plagiarism detection. Plagiarism refers to the pilfering of another person's writing into one's piece of work, which can be an essay, assignment, or research paper without attributing the original writer. The primary expectation in every class is that whatever you write will be your own words, generated from your understanding. Plagiarists perform this unacceptable activity and pass the respective stage without being caught. It becomes very tiring for the professors or checkers to identify the possible cribbing involved. PlagiDetect is a solution to the problem of plagiarism. It will provide the facility of detecting plagiarism in the text with flexible options of accessing text files. This project focuses on building a web application that performs plagiarism detection among text files. At present, it successfully processes text files, aiming to process files of other formats like pdfs, and documents in the near future.

The main objective of this project is to create a web application to detect plagiarism. The registered user should receive a detailed report of the result that PlagiDetect will generate. The end users will have to register to PlagiDetect by creating an account. Users will be able to give access to the text files with four options: upload the text files from any location on the computer into PlagiDetect web application, enter a particular drive name in which all the available text files will be processed, enter a particular path to a folder name, and to check whole of the local computer for the text files. In this way, a professor can check the students' assignments in a fraction of a second. Plagiarism detection is a great way to check the originality of a document. The results of each plagiarism detection should be read and understood to address the shortcomings in the composition.

SYSTEM STUDY AND ANALYSIS

COLLECTING AND INTERPRETING FACTS

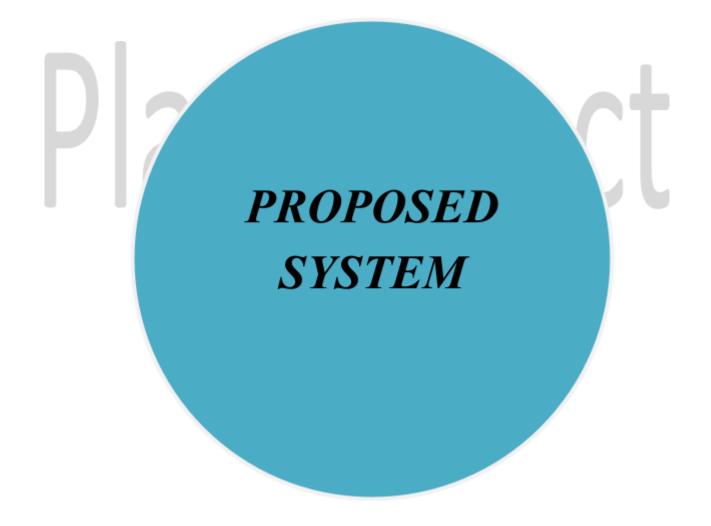
Plagiarism detection services are a powerful tool to help encourage academic integrity. One of the most important parts of that job is performing plagiarism analyses. These are done for a variety of reasons including providing expert witness testimony for court cases, verifying authenticity of a large project, checking a website before sale or even examining a book before publication. There are two kinds of plagiarism analyses. The first is when you're comparing two or more known works against each other and the second is when you're trying to determine the originality of an unknown work by comparing it against as much of the world you can. Turnitin is a program that compares your students' submissions to information on the internet and other papers that have been submitted to Turnitin. Turnitin is a tool that students and instructors can use to identify potential instances of plagiarism. When you submit your paper to Turnitin, you will get a similarity report that should be reviewed carefully so you can make any needed revisions to your paper.

Grammarly's algorithms flag potential issues in the text and make context-specific suggestions to help with grammar, spelling and usage, wordiness, style, punctuation, and even plagiarism. The software explains the reasoning behind each suggestion, so you can make an informed decision about whether, and how, to correct an issue. Underlying all of Grammarly's products is a sophisticated artificial intelligence system built to analyze sentences written in English. Grammarly's team of computational linguists and deep learning engineers designs cutting-edge algorithms that learn the rules and hidden patterns of good writing by analyzing millions of sentences from research corpora. (A corpus is a large collection of text that has been organized and labeled for research and development purposes.)

IDENTIFYING THE PROBLEMS

Adoption of these services has proven to be controversial due to ethical concerns about students' rights. Central to these concerns is the fact that most such systems make permanent archives of student work to be re-used in plagiarism detection. This computerization and automation of plagiarism detection is changing the relationships of trust and responsibility between students, educators, educational institutions, and private corporations. Educators must respect student privacy rights when implementing such systems. Student work is personal information, not the property of the educator or institution. The student has the right to be fully informed about how plagiarism detection works, and the fact that their work will be permanently archived as a result. Furthermore, plagiarism detection should not be used if the permanent archiving of a student's work may expose him or her to future harm.

Moreover, the users only have a single option of uploading their files to the software's web, mobile, or desktop application. There is no other option to give access to the files. For instance, if a professor has kept all the students' assignments in a specific folder, then it will be less time-consuming for him if he just types the path to the files instead of uploading them. By just typing the path, he saves a lot of time and the rest of the work should be done by the software. If the software reports a high number, editors or professors might unjustly consider a submission as unequivocal plagiarism. Universities formally define 'acceptable' levels of plagiarism, evaluated by the software, for various degree levels. Teachers want the software to flag up the 'bad' papers, so they don't have to read them. But students, afraid of having accidentally plagiarized, use the same systems to rewrite their work, swapping words with synonyms and rearranging sentences until the number looks good, to the detriment of readability.



MAIN FEATURES

1 – FLEXIBILITY IN ACCESSING TEXT FILES

PlagiDetect allows the end users to choose from four options to access the text files. Let us discuss these four choices one by one as follows:

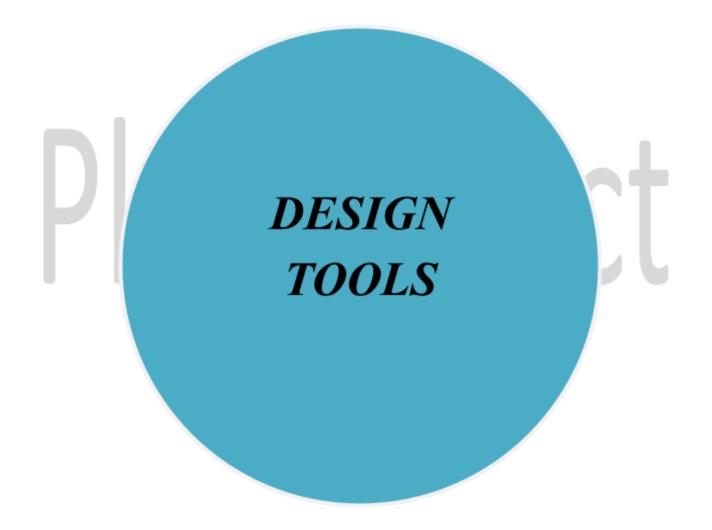
- Upload the text files The first option allows the end users to upload the text files. It restricts all other formats of files like .pdf, .png, or .jpg. Of course, in the coming future, PlagiDetect will be capable of uploading more flexible file formats. It uploads these files and saves them in the database. Then, it performs plagiarism detection among these files and saves the result in a file, which is then emailed at the registered email.
- Access all the text files from the local computer (all drives) The second option allows the users to grant access to all the text files in the local computer's drives (be it a pen drive) at one click. It then performs backend logic to calculate similarity among those text files and creates as many text files of outputs as there are number of drives in the local computer.
- Access the text files from a particular drive The third option asks the end user to enter the name of any drive the user wishes to give to allow access to the text files within that drive. It does not takes the text files within any folders in the drive yet, but all the text files present immediately in the drive. After submitting just the drive name, the web application performs plagiarism detection on all those text files, generates an output file, and finally sends as a mail.
- Access the text files from a specific folder (a path) The last but not the least option allows the users to enter any path to the saved text files. The path can contain one or more than one folders within. Again, the text files are searched within the path and

processed to produce a result telling the similarity percentage among the selected text files. PlagiDetect is on its way to add more options and more features in the near future.

2 – PERSONALIZED RESULT

After selecting the option of choice, the web application – PlagiDetect performs the logical operations on the accessed text files. The plagiarism detection report or result is not shown on the screen, instead it is sent at the registered email-id of the user. This ensures privacy, hence security of the plagiarism result. The user will get the output in a text file. In the above mentioned options, all options except one outputs a single output file. The second option, that is, 'Access all the text files from the local computer (all drives)', generates as many output files as the number of drives in the computer. For example, if there are four drives (C:,D:,E:,G:), then four output files, making it clear from their names as to which drive's output it contains will be generated and send to the registered email-id.

The output files are also generated in the specific places from where the text files have been accessed except for the first option of uploading the text files. In the second option, an output file will be created in every drive of the local computer. Therefore, the user can check the output in the particular drives as well. Also, if the user again chooses the same option, these files will not be overwritten; instead it will rewrite the files. In case of just one drive, the output file will be created in that drive only. Further, if the last option is chosen, then in addition to mailing the output file to the registered email-id, PlagiDetect also creates it in that folder for the user. Still comprising unique features, we are working on enhancing this web application even more. Features like accessing text files from the local area network (LAN), uploading files of more formats, allowing Hindi and Punjabi texts for plagiarism detection, and many more are on their way.



FRONTEND TOOLS

HTML (Hyper Text Markup Language) - First developed by Tim Berners-Lee in 1990, HTML is short for Hypertext Markup Language. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another. In this web application, templates are created with the help of HTML source code.

CSS (Cascading Style Sheets) - Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. PlagiDetect uses CSS to style the templates written in HTML, which gives users a wonderful experience.

JS (JavaScript) – It is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. 97% it client-side for web Over of websites use page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on the user's device. PlagiDetect also makes use of JS to perform some restrictions while uploading files in the first option.

BACKEND TOOLS

PYTHON - Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- System scripting.

As can be seen from the various uses of python, web development (server-side) is used in our project – PlagiDetect. At the backend, when the text files are to be checked for plagiarism against one another, Python logics and functionality helps a lot. Various inbuilt modules and classes in Python work wonders at vital steps in this project. The following modules and classes are used in the process of making of this web application.

- os The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules.
- re A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern.
- difflib (SequenceMatcher) SequenceMatcher is a class available in python module named "difflib". It can be used for comparing pairs of input sequences.
- pathlib (Path) It provides various classes representing file system paths with semantics appropriate for different operating systems.

BOTH FRONTEND AND BACKEND

DJANGO - Django is an open source web framework used for rapid development, pragmatic, maintainable, clean design and secure websites. The main goal of this is to allow developers to focus on components of the application that are new instead of spending time on already developed components. Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Django functionality also includes sending emails. To do that, Django use smtplib module that is provided by Python. There are several ways to implement email sending in Django:

- 1. Use light thin wrapper function that Django provide such as send_mail(), send_mass_mail(), and mail_admins()
- 2. Create a class that extends EmailMessage class, or use EmailMessage class instead.

Simple Mail Transfer Protocol is a standard protocol that is being used for sending emails, one of killer application that brought internet to the spotlight. It mainly uses TLS (Transfer Layer Security) or SSL (Secure Sockets Layer) as its security protocol and port 587 as its standard port, according to RFC 2476 in 1999. SSL and TLS, with the later is the recent one, are a way for web client and web server to authenticate each other, either by adding SSL certificate in SSL, or adding encryption algorithm and separation of handshake and record protocol in TLS.

WORKING METHODOLOGY

AGILE METHODOLOGY

Agile methodology is the ideal approach for the PlagiDetect application. It focuses on building a single feature at a time. Agile uses sprints or a set of times when a particular trait of the application should be developed, tested, and presented. It incorporates the entire Software Development Life Cycle (SDLC) for a feature into each sprint. It helps to stick to a planned schedule but also allows for repeated analyses. The broad idea is to split the entire project into multiple features (four in this case) through which the same set of steps follow. This approach is known as **iterative and incremental development**. The increment happens from one component to another, and repeated execution is performed iteratively for each part or module. These modules further comprise sub-modules. At every level, a similar approach follows. Let us discuss the incremental and iterative approach separately as well as a combination to have a deeper understanding:

Incremental approach – The incremental approach is based on the principle that those involved in a project should at the outset focus on the key business objectives that the project is to achieve and be willing to suspend detailed consideration of the minutiae of a selected solution.

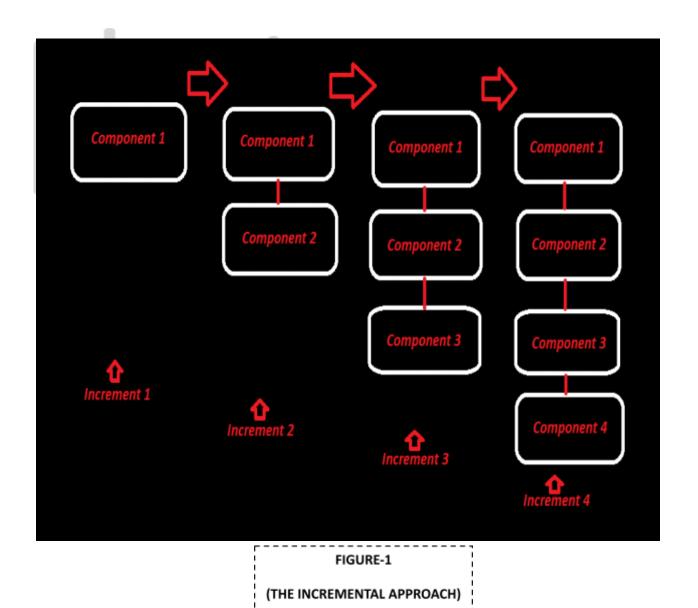
Iterative approach – An iterative approach is one where the content of the discussion, stimulus, or sometimes even the methodology is adapted over the course of the research programme. Learning from initial research sessions is used to influence the inputs for subsequent interviews. For example, in a creative development project the ad agency will amend scripts and storyboards overnight based on early feedback.

Iterative and Incremental approach (combination) - The combination of the two approaches: iterative and incremental has been long-standing and has been widely suggested for large development efforts.

START ENTER NAME AND EMAIL-ID CHOOSE FROM THE FOLLOWING OPTIONS: 1 - UPLOAD TEXT FILES 2 – ALLOW PlagiDetect TO ACCESS ALL THE TEXT FILES OF THE LOCAL COMPUTER 3 - ALLOW PlagiDetect TO ACCESS THE TEXT FILES FROM A **PARTICULAR DRIVE** 4 - ALLOW PlagiDetect TO ACCESS THE TEXT FILES FROM A PARTICULAR FOLDER IN A DRIVE NO NO NO OPTION 1 OPTION 2 OPTION 3 OPTION ? 4? YES YES YES YES **Drag and Drop** Access text files Access all the text Access text files according to a text files files of the local from a particular path particular drive computer K PLAGIARISM AMONG ALL THE TEXT FILES, VERATE OUTPUT FILE(S), AND EMAIL THEM IN THE REGISTERED ID FLOWCHART-1 **STOP** (WORKING OF PlagiDetect)

INCREMENTAL APPROACH

Let us look at the Incremental approach. The development is broken down into four modules. The four modules are the four components of the product. The component that needs to re-build entirely is taken as the first component. Therefore, it undergoes all the steps starting from requirements up to review. The other three components go through only the four steps, that is, design, testing, implementation and review. Together, all the four components are integrated to design the final product. The additional two steps, that is, deployment and maintenance is performed at the end.



INCREMENTAL APPROACH IN PlagiDetect

In our Django web application – PlagiDetect, the components will be the following four options:

- 1 UPLOAD TEXT FILES
- 2 ALLOW PlagiDetect TO ACCESS ALL THE TEXT FILES OF THE LOCAL COMPUTER
- 3 ALLOW PlagiDetect TO ACCESS THE TEXT FILES FROM A PARTICULAR DRIVE
- **4** ALLOW PlagiDetect TO ACCESS THE TEXT FILES FROM A PARTICULAR FOLDER IN A DRIVE

It can be clearly seen from the FIGURE-1 (THE INCREMENTAL APPROACH) that the components in this approach are incremented at the next stage. If just this approach is followed, then every minor and major functionality is performed one by one with every component. In our project, that would mean writing standalone multiple modules instead of a faster approach. In this approach, the system is put into production when the first increment is delivered. The first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments. Once the core product is analyzed by the client, there is plan development for the next increment. This approach is used when requirements of the system are clearly understood, when demand for an early release of a product arises, when software engineering team are not very well skilled or trained, when high-risk features and goals are involved, and for web application and product based companies. Errors are easily identified in this approach. This model is less costly as compared to other approaches. It is much more flexible than others in terms of change in requirements and scope.

ITERATIVE APPROACH

The Iterative approach will comprise of four iterations; one for each component. The steps involved in these iterations will include design, testing, implementation and review. These four steps are the common steps for all the components. The first component undergoes additional two steps at the initial stage, that is, requirements and analysis. The final product, which is the integration of all the four components, will undergo all the eight steps.

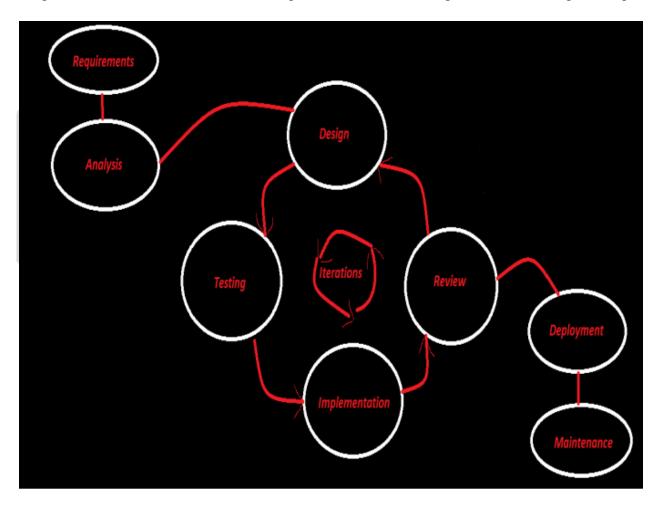


FIGURE-2 (THE ITERATIVE APPROACH)

ITERATIVE APPROACH IN PlagiDetect

In this approach, there are multiple fixed iterations for each component (options in this case). Unlike the more traditional waterfall model, which focuses on a stringent step-by-step process of development stages, the iterative model is best thought of as a cyclical process. Enhancements can quickly be recognized and implemented throughout each iteration, allowing the next iteration to be at least marginally better than the last.

- Planning & Requirements: As with most any development project, the first step is go
 through an initial planning stage to map out the specification documents, establish
 software or hardware requirements, and generally prepare for the upcoming stages of the
 cycle.
- Analysis & Design: Once planning is complete, an analysis is performed to nail down
 the appropriate business logic, database models, and the like that will be required at this
 stage in the project.
- Implementation: With the planning and analysis out of the way, the actual implementation and coding process can now begin.
- **Testing**: Once this current build iteration has been coded and implemented, the next step is to go through a series of testing procedures to identify and locate any potential bugs or issues that have cropped up.
- Evaluation: Once all prior stages have been completed, it is time for a thorough evaluation of development up to this stage.

ITERATIVE AND INCREMENTAL APPROACH

Agile approach incorporates the philosophy of iterative and incremental software development that is modeled around a gradual increase in feature additions and a cyclical release and upgrade pattern. The outcome of the subsequent iteration is an enhanced working increment of the product. This is repeated until the product accomplishes the required functionalities.

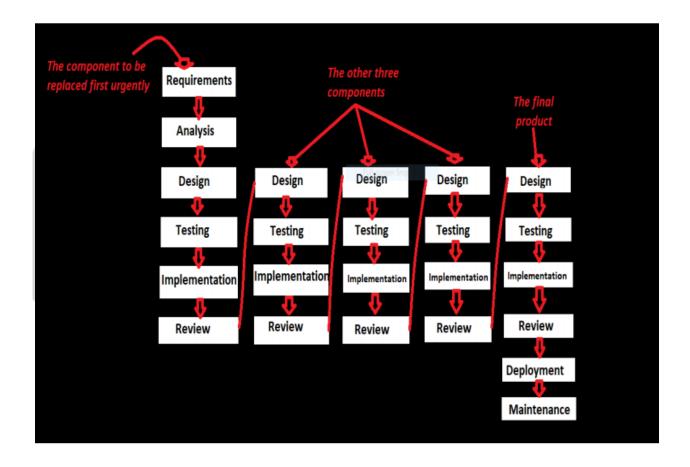
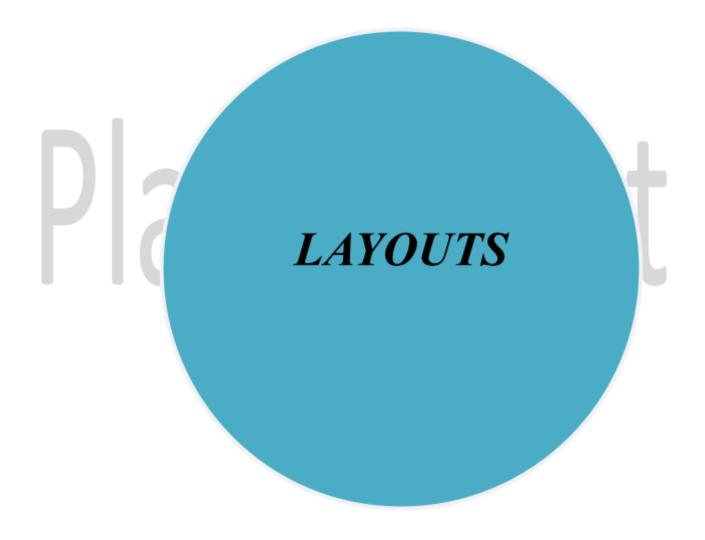


FIGURE-3 (THE ITERATIVE AND INCREMENTAL APPROACH)

ITERATIVE AND INCREMENTAL APPROACH IN PlagiDetect

In PlagiDetect, we have implemented Agile methodology (both iterative and incremental approach simultaneously). Agile allows using the combination of both incremental and iterative approach. It results in a faster methodology and offers the following advantages:

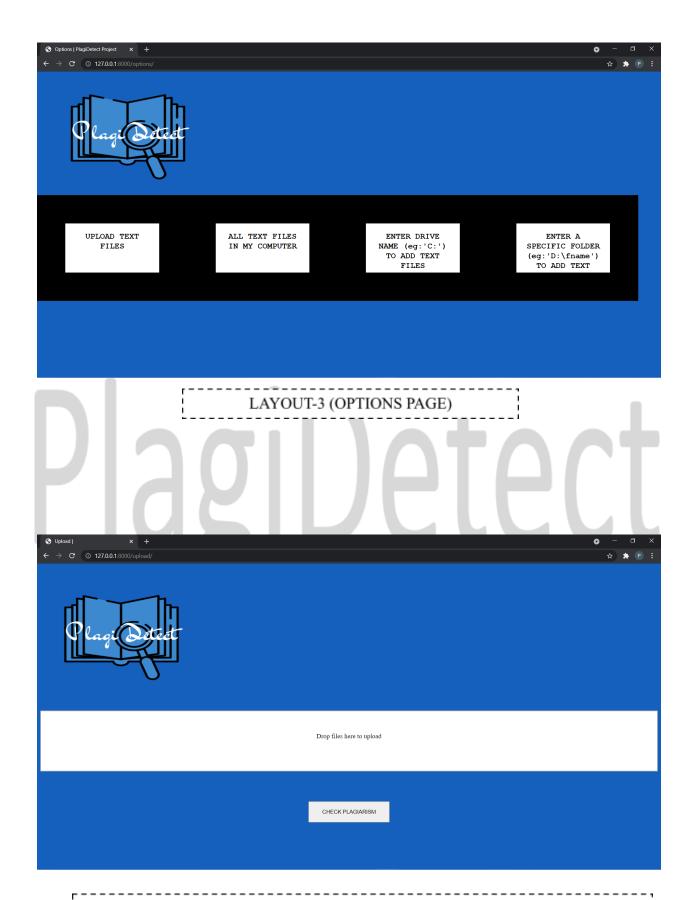
- **1. Improved quality-**When using an agile methodology, teams can breakdown projects into **sprints** and **collaborate** with one another to **provide high-quality results**. The fast processing of PlagiDetect will automatically enhance its quality in the market.
- **2. Speed and flexibility-**The second benefit of using agile is its speed and flexibility thanks to **Scrum framework**. This practice places change at the heart of its development. PlagiDetect performs backend operations faster as compared to when single approach is utilized. It also allows the making of this project faster and smartly.
- 3. Complete visibility of the progress of each project in real-time-Another advantage of using an agile approach is the transparency of each project thanks to frequent exchanges with clients. This allows them to feel more involved and ask for changes throughout the project. During the project building, we were able to run each option separately regardless of the other options.
- **4. Stakeholders engagement-**A key part of using an agile method is the **involvement of stakeholders** when completing projects.
- **5. Cost control**-An agile method can also be used to **improve cost control**. After each stage, the team reviews the **budget** when making future decisions. Then, they decide if they will continue, suspend or cancel tasks or even the project itself.



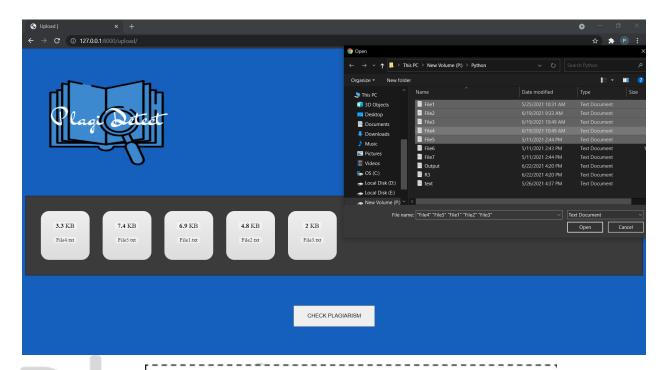
LAYOUTS



LAYOUT-2 (REGISTRATION PAGE WITH VALIDATIONS)

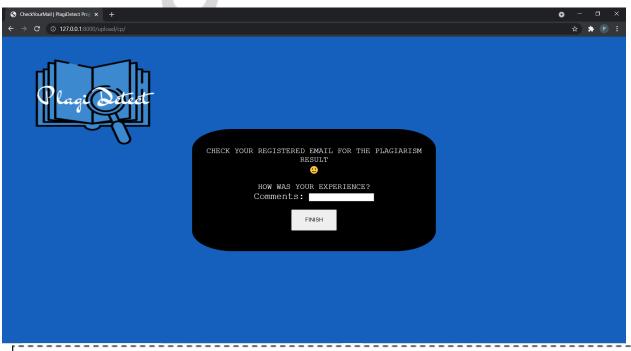


LAYOUT-4 (FIRST OPTION – DRAG AND DROP/UPLOAD TEXT FILES)

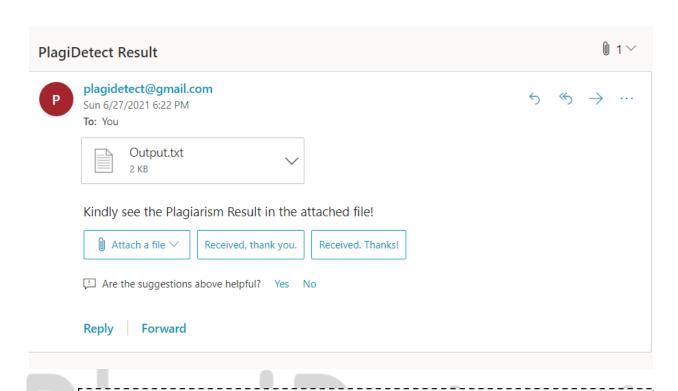


LAYOUT-5 (UPLOADING TEXT FILES)

Plagibether

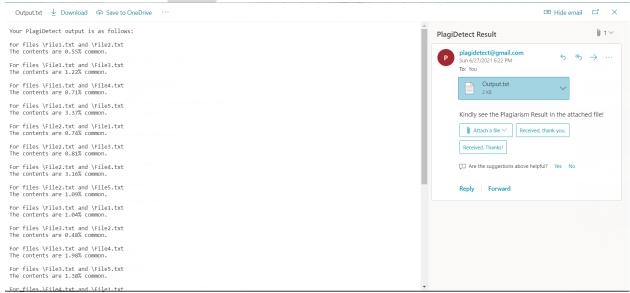


LAYOUT-6 (REDIRECT PAGE FOR ALL OPTIONS TELLING TO CHECK MAIL)

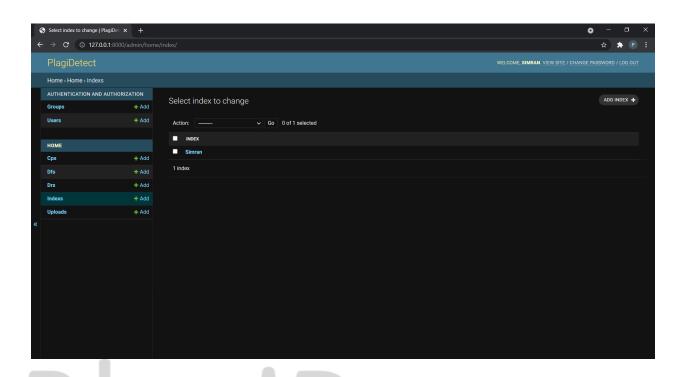


LAYOUT-7 (RECEIVED EMAIL FOR THE FIRST OPTION)

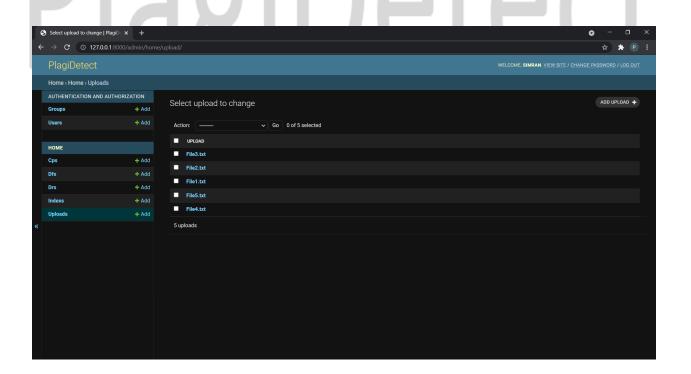
Iagille Tect



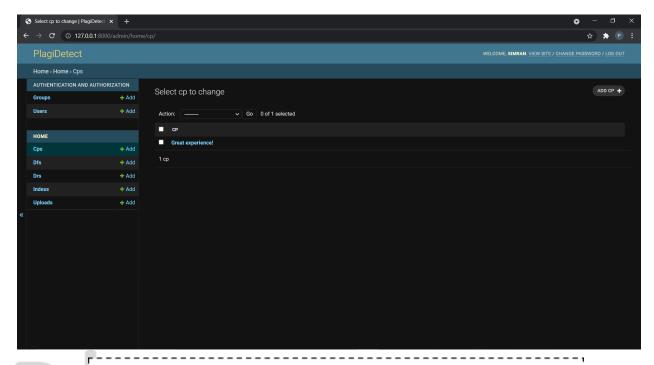
LAYOUT-8 (RECEIVED OUTPUT.TXT FILE IN EMAIL CONTAINING THE RESULT)



LAYOUT-9 (REGISTRATION INFORMATION STORED IN DATABASE)



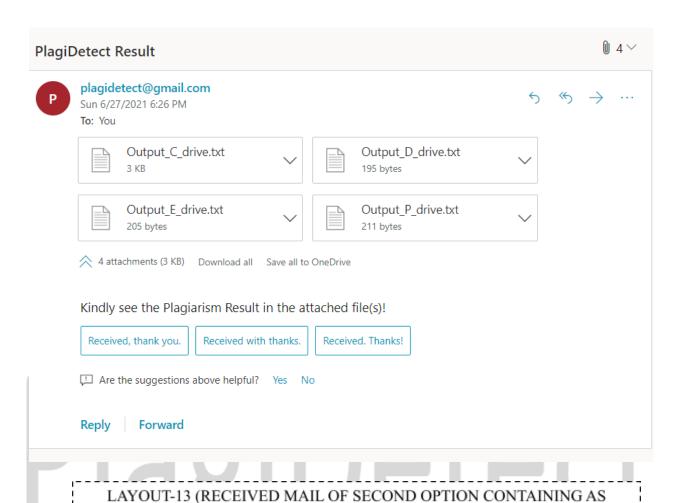
LAYOUT-10 (UPLOADED FILES STORED IN DATABASE)

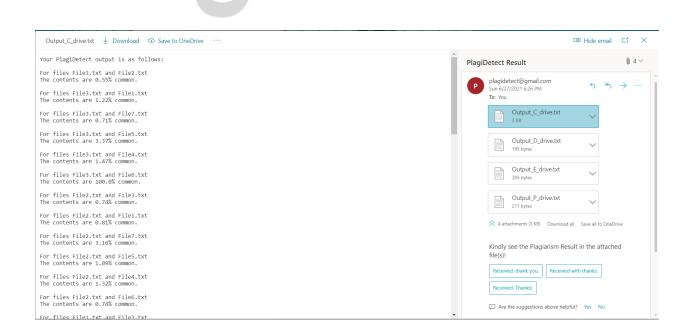


LAYOUT-11 (COMMENTS STORED IN DATABASE)



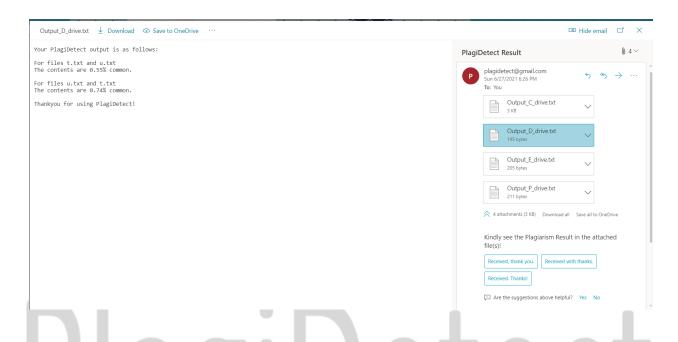
LAYOUT-12 (WEB PAGE FOR SECOND OPTION TAKING PERMISSION TO FETCH TEXT FILES FROM ALL THE DRIVES)





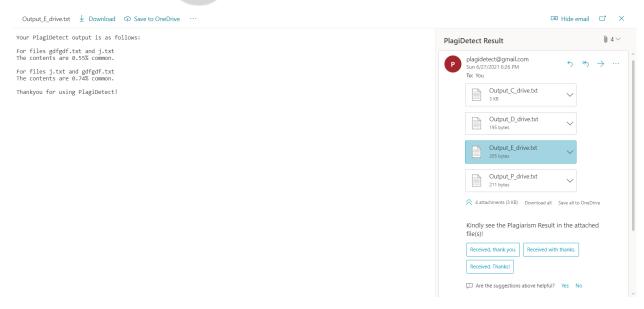
MANY OUTPUT FILES CLEARLY MENTIONING THE DRIVE NAME)

LAYOUT-14 (DETAILED OUTPUT OF FIRST DRIVE IN SECOND OPTION)

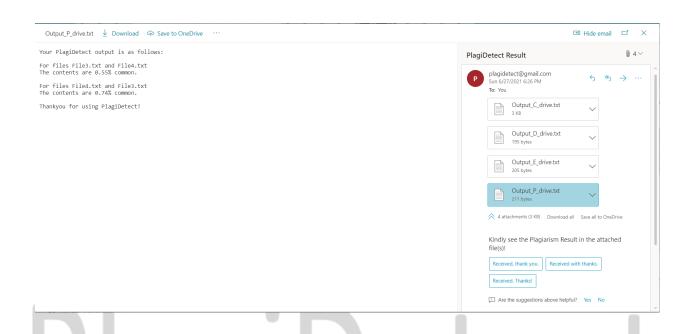


LAYOUT-15 (DETAILED OUTPUT OF SECOND DRIVE IN SECOND OPTION)

Iagiveleu



LAYOUT-16 (DETAILED OUTPUT OF THIRD DRIVE IN SECOND OPTION)

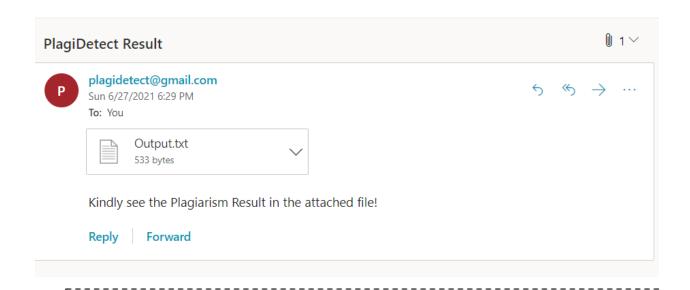


LAYOUT-17 (DETAILED OUTPUT OF LAST DRIVE IN SECOND OPTION)

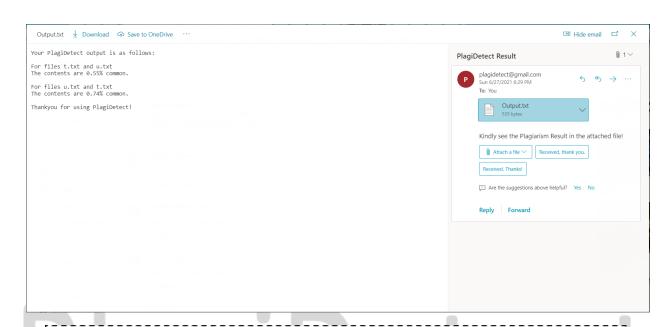


LAYOUT-18 (WEB PAGE FOR THIRD OPTION)

ENTER THE NAME OF THE DRIVE (eg - C:)	
DriveName: D:	
Submit	
LAYOUT-19 (EXAMPLE INPUT FOR THIRD OPTION	



LAYOUT-20 (RECEIVED EMAIL FOR THIRD OPTION)



LAYOUT-21 (DETAILED OUTPUT FOR THIRD OPTION)

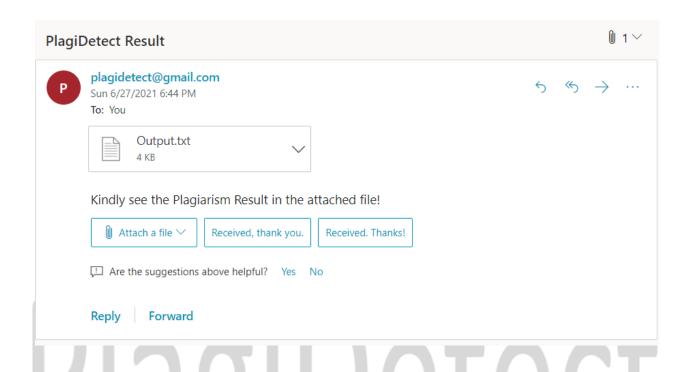
Idgil/Cici

ENTER THE PATH
(eg - D:\folder_name OR c:\fol1\fol2..)

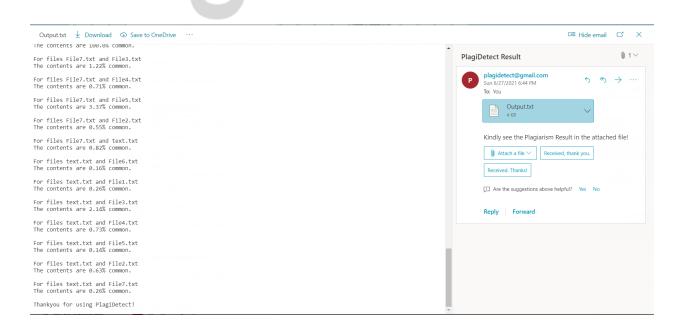
Path: P:\Python

Submit

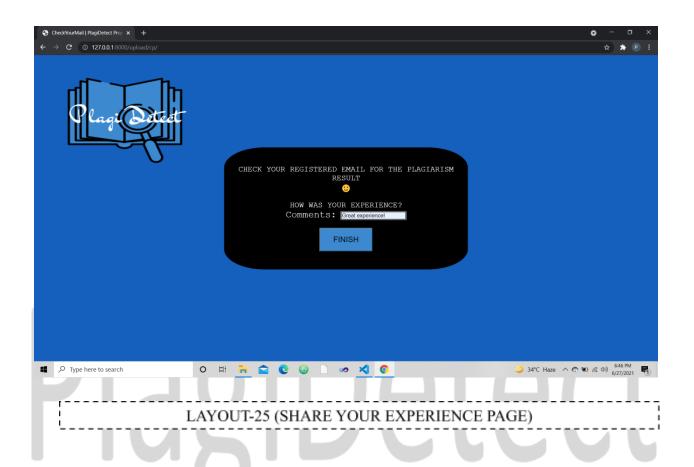
LAYOUT-22 (EXAMPLE INPUT FOR FOURTH OPTION)

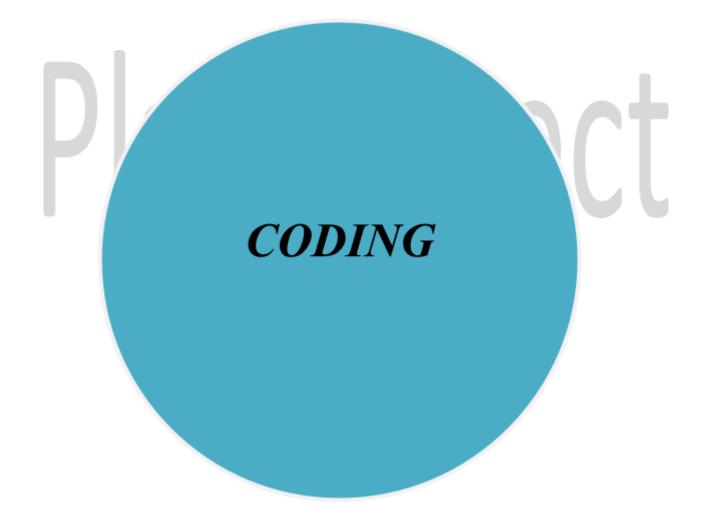


LAYOUT-23 (RECEIVED EMAIL FOR FOURTH OPTION)



LAYOUT-24 (DETAILED OUTPUT FOR FOURTH OPTION)





CODING

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

    CODE-1 (INSTALLED APPS IN DJANGO SETTINGS.PY)
```

Django provides some pre-installed apps for users. To see pre-installed apps, navigate to projectName -> projectName -> settings.py

In your settings.py file, you will find INSTALLED_APPS. Apps listed in INSTALLED_APPS are provided by Django for developers comfort. There is a need to add our own application in this list, which is present in settings.py. The name of our application is 'home', which is added in the end followed by a comma. This step is very important and basic in nature. Only after this addition will our django web application move ahead in other aspects. This also is the pre-step of running makemigrations and migrate commands in django terminal.

```
from django.contrib import admin
from django.urls import path, include
```

CODE-2 is the urls.py inside the 'PlagiDetect' (project_name) directory. The first path in the urlpatterns is the admin path, which directly takes to the admin interface. It takes the username and the password from the administrator of the django web application. By default, it has the django specific headings written on the admin interface. These can be changed by setting the three parameters mentioned in the above code: admin.site.site_header, admin.site.index_title, and admin.site.site_title. For our project, we have set these parameters to "PlagiDetect", "Welcome to PlagiDetect", and "PlagiDetect Project". The second path in the urls patterns list redirects to another urls.py file, which is located inside the 'home' folder. In the first two lines, admin, path, and include are imported. This import is very important to perform as the code will throw an error if these are used without importing them.

```
from django.contrib import admin
from django.urls import path
from home import views
from django.conf import settings
from django.conf.urls.static import static
from home.views import MainView, Welcome, lc, drive, df
from django.contrib.staticfiles.urls import staticfiles urlpatterns
urlpatterns = [
    path('',views.Welcome.as view(),name = 'welcome'),
    path('index/', views.index, name = 'index'),
    path('options/', views.options, name = 'options'),
    path('upload/',MainView.as view(),name = 'main-view'),
    path('upload/fileupload/',views.file upload view,name = 'fileupload'),
    path('plag/',views.plag,name = 'plag'),
    path('upload/cp/',views.cp,name = 'cp'),
    path('lc/',lc.as view(),name = 'lc'),
    path('lc/lcc/', views.lcc, name = 'lcc'),
    path('drive/',views.drivee,name = 'drive'),
    path('drivee/',views.drivee,name = 'drivee'),
    path('df/',views.dff,name = 'df'),
    path('dff/',views.dff,name = 'dff'),
   # ... more URL patterns here
if settings.DEBUG:
    urlpatterns += static(settings.MEDIA URL, document root=settings.MEDIA ROOT)
    urlpatterns += static(settings.STATIC URL, document root=settings.STATIC ROOT)
            CODE-3 (REDIRECTED URLS OF OUR WEB APPLICATION)
```

CODE-4 is a model class in models.py. This is a model form, which is created to set the field names to be stored in the database of the django admin. As we can see, every database field has a datatype which specifies the type of data it will take from the end user. In this class, there are two fields: name and email. Both the name and the email field are set to CharField with max_length as 122. The date field does not take any user input; it only saves the first two fields with the current data and time. The __str__ function inside this class takes 'self' as the only argument; its only task here is to save the fields with the name of the field mentioned in the parantheses in the return statement of this function.

```
class Index(models.Model):
    name = models.CharField(default = "", max_length=122)
    email = models.CharField(default = "", max_length=122)
    date = models.DateField()
    class Meta:
        db_table = "index"

def __str__(self):
        return str(self.name)
```

CODE-4 (ONE OF THE MODELS – REGISTRATION FORM)

CODE-5 shows two of the many forms used in this web application in forms.py. Firstly, forms are imported from django, and secondly, the ModelForms are imported from the models.py. In this, we mention the field names that we wish to include from the models made in models.py. As can be seen, in every class a meta class is created which specifies the name of the model and the fields used.

```
from django import forms
from .models import Index, Upload, DF, Dr, CP

class IIndex(forms.ModelForm):
    class Meta:
        model = Index
        fields = ["name","email"]

class UUpload(forms.ModelForm):
    class Meta:
        model = Upload
        fields = ["upload"]
```

CODE-5 (TWO OF THE FORM CLASSES – REGISTRATION AND FILE UPLOAD)

CODE-6 is the main function which checks the similarity among text files that are accessed by either of the four options available to the end user. It is creating a file named 'Output.txt' in which the final output of either of the options is saved and send. It takes three arguments: FC – It is a dictionary containing the file names as keys, and the file content as values, NL – It is new list comprising only of the content of the files without any escape sequences, and FN – It is a list of file names only.

```
def sim(FC,NL,FN):
    with open("Output.txt", "w") as o:
        o.write("Your PlagiDetect output is as follows:")
        o.write("\n\n")
        for j in range(len(list(FC.values()))):
            for k in range(len(list(FC.values()))):
                if(j != k):
                    s = SequenceMatcher(None, NL[j], NL[k])
                    sim = s.ratio()
                    sim = sim*100
                                                                  1
                    sim = round(sim, 2)
                    o.write(f"For files {FN[j]} and {FN[k]}")
                    o.write("\n")
                    o.write(f"The contents are {sim}% common.")
                    o.write("\n\n")
        o.write("Thankyou for using PlagiDetect!")
        global out
        def out(request):
            return('Output.txt')
```

CODE-6 (SIMILARITY CHECKING FUNCTION)

CODE-7 is a function of uploading multiple files to the django database. This function is connected with the template of html dealing with the file uploads. Also, it is linked with the urls in urls.py. The action specified in the action parameter of the form tag in the html template decides which function in views.py will run. This function runs only if the method specified in the method parameter of the file upload html template is "POST". Otherwise, it will throw an error. Here my_files is an object, and request.FILES.get('file') tells that it will allow multiple files to get uploaded. Here 'file' in the brackets is the value of the 'name' parameter in the form tag inside the file upload html template.

```
def file_upload_view(request):
   if request.method == 'POST':
      my_file = request.FILES.get('file')
      Upload.objects.create(upload = my_file)
      return HttpResponse('')
```

CODE-7 (FUNCTION RESPONSIBLE FOR UPLOADING FILES)

CODE-8 is a function in django's views.py. This function is specifically created to handle sending of output files via emails at the registered email id's. id(request) is a global function that is created in another function where email is directly used. Therefore, in order to extract different email-id's from that function everytime, a global function is created and is used or call in this function to use the dynamic values of email field in the registration form. Python has EmailMessage class which can be used build email messages. This class ahs the required methods to customize different parts of the email message like - the TO and FROM tags, the Subject Line as well as the content of the email.

Habble Company of the Company of the

```
def mail(request):
    email = id(request)
    html_content = "Kindly see the Plagiarism Result in the attached file!"
    email = EmailMessage("PlagiDetect Result", html_content, "plagidetect@gmail.com", [email])
    email.content_subtype = "html"
    email.attach('Output.txt',OpenRead('Output.txt'),'text/plain')
    email.send()
```

CODE-8 (FUNCTION RESPONSIBLE FOR SENDING MAIL)

CODE-9 is one of the functions in views.py. This function is taking data from the end user in the registration form. Besides taking and saving data to the database, this function also is validating the form fields. If the user clicks on the submit button without filling any one of the fields, then a validation is thrown as an error message as was seen in one of the layouts.

```
def index(request):
    if request.method == "POST":
        form = IIndex(request.POST)
        if form.is_valid():
            try:
                name = form.cleaned_data['name']
                email = form.cleaned_data['email']
                global id
                def id(request):
                    return email
                index = Index(name = name, email = email, date = datetime.today())
                index.save()
                return HttpResponseRedirect('/options')
            except:
                pass
    else:
        form = IIndex()
    return render(request, 'docs/index.html',{'form':form})
```

CODE-9 (FUNCTION RESPONSIBLE FOR VALIDATING REGISTRATION FORM)

CODE-10 is a template of html for the options web page for accessing files. In this, the user can choose any one of the four available options: 'Upload Text Files', 'All Text Files In My Computer', 'Enter Drive Name (eg:'C:') To Add Text Files', and 'Enter A Specific Folder (eg:'D:\fname') To Add Text Files'. Any option chosen by the user takes to the respective redirected path of the html template.

```
{% extends 'base.html' %}
{% block title %}Options |{% endblock title %}
{% block body %}
   <div id="div1">
       <a href = "/upload">UPLOAD TEXT FILES</a>
           <a href = "/lc">ALL TEXT FILES IN MY COMPUTER</a>
           <a href = "/drive">ENTER DRIVE NAME (eg: 'C:') TO ADD TEXT FILES</a>
           <a href = "/df">ENTER A SPECIFIC FOLDER <br/>br>(eg:'D:\fname') <br>TO ADD TEXT FILES</a>
           ⟨/li⟩
       </div>
{% endblock body %}
```

CODE-10 (HTML TEMPLATE FOR OPTIONS WEB PAGE)

CODE-11 is the template code of html for the file upload web page. It extends the master or the base template in the very beginning, which remains the same for all the other templates. One can create more than one master or base templates as well if the functionality demands for it. In PlagiDetect, we created two master or base templates. One base template named b.html was created solely for file upload web page — upload.html. The second base template named base.html is the common master template for all the other templates. This is done because the functionality of the file upload template demanded some additional links that only this template needed.

```
{% extends 'b.html' %}

{% load static %}

{% block title %}Upload |{% endblock title %}

{% block content %}

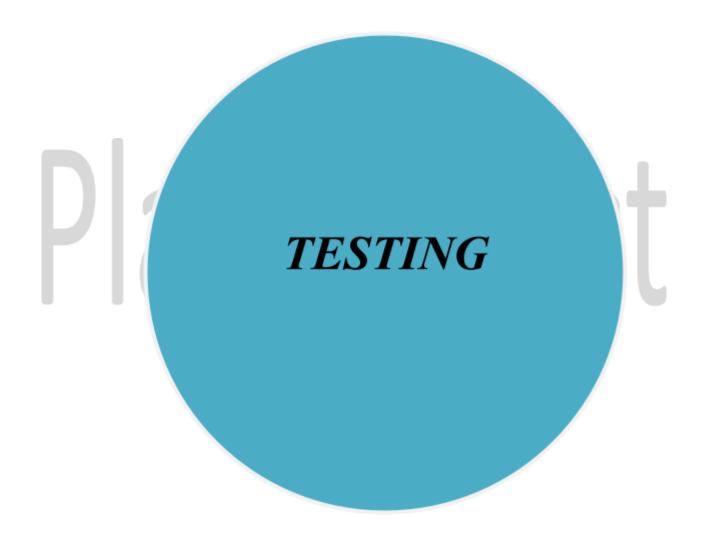
<form action="fileupload/" method = "POST" class="dropzone dz" id = "special" name = "form_type">
{% csrf_token %}

<div id = "divv" class="fallback">

</div>
</form>
<form action = "/plag/" method = "POST" id = "special1" name = "form_type">
{% csrf_token %}

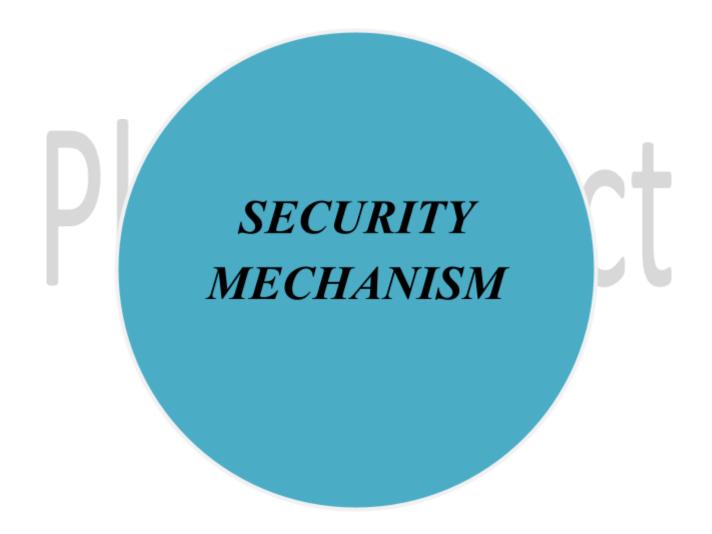
<br/>
<br/>
<br/>
<br/>
<br/>
<br/>
<input id = "button" type="submit" value="CHECK PLAGIARISM"></center>
</form>
{% endblock content %}
```

CODE-11 (HTML TEMPLATE FOR FILE UPLOAD WEB PAGE)



Software testing life cycle contains the following steps:

- 1. Requirement Analysis The first step of the manual testing procedure is requirement analysis. In PlagiDetect, the requirement of the four options is tested by testing all the four options for multiple number of files, and so on.
- 2. Test Plan Creation Test plan creation is the crucial phase of STLC where all the testing strategies are defined. Tester determines the estimated effort and cost of the entire project. The cost in PlagiDetect is in terms of time and speed.
- 3. Environment setup Setup of the test environment is an independent activity and can be started along with **Test Case Development**. This is an essential part of the manual testing procedure as without environment testing is not possible. In PlagiDetect, the number of drives were varied to change the environment for testing.
- <u>4. Test case Execution</u> Test case Execution takes place after the successful completion of test planning. In PlagiDetect, various test cases were executed at every point in every option for multiple cases and scenarios.
- 5. Defect Logging Testers and developers evaluate the completion criteria of the software based on test coverage, quality, time consumption, cost, and critical business objectives. There are many features that will be added to reduce the drawbacks of PlagiDetect.
- <u>6. Test Cycle Closure</u> The test cycle closure report includes all the documentation related to software design, development, testing results, and defect reports.



The primary benefits of using a secure Software Development Life Cycle (SDLC) include:

• Early identification of vulnerabilities in the application security.

Vulnerabilities like uploading of other file formats in the name of .txt files, and so on, are possible in the web application of PlagiDetect.

More secure software as security is a continuous concern.

Maintaining and updating the security policies and loopholes as new features are continuously added in this web application is a must.

• Stakeholders are aware of the security risks in real-time.

Security mechanisms and updating of the security policies and the algorithms are equally important because stakeholders are aware of the risks involved in real – time running application.

Reduced cost, time, and effort to mitigate security risks as they are detected early in the SDLC.

Detection of security risks was done in earlier phases of PlagiDetect as the project was building. This reduced the cost, time, and the effort to eradicate some of the security threats, which were taken care of at the right time.

An overall reduction in business risks for the enterprise.

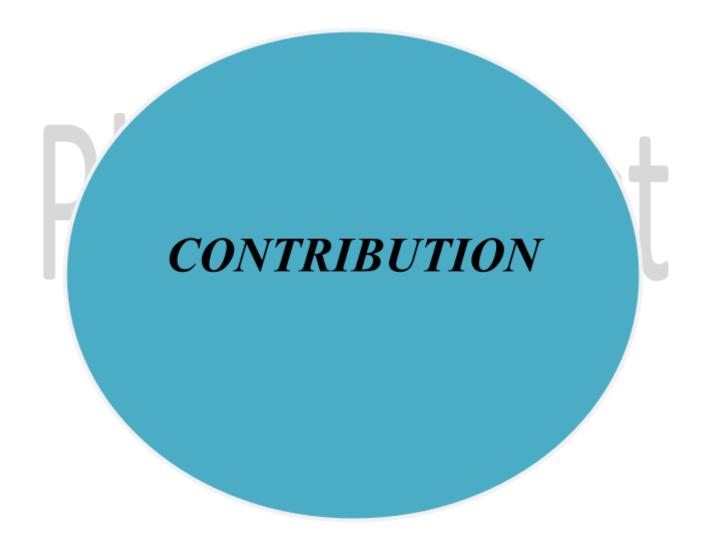
Earlier detection, prevention, and mitigation of the security risks in PlagiDetect allowed us to reduce the overall business risks involved during the future deployment of the django web application.

IMPLEMENTATION AND MAINTENANCE

MAINTENANCE AND UPDATIONS IN THE FUTURE

Maintenance includes all the activity after the installation of software that is performed to keep the system operational. As we have mentioned earlier, software often has design faults. The two major forms of maintenance activities are adaptive maintenance and corrective maintenance. It is generally agreed that for large systems, removing all the faults before delivery is extremely difficult and faults will be discovered long after the system is installed. As these faults are detected, they have to be removed. Maintenance activities related to fixing of errors fall under corrective maintenance. Removing errors is one of the activities of maintenance. Maintenance also needed due to a change in the environment or the requirements of the system. The introduction of a software system affects the work environment. This change in environment often changes what is desired from the system. Furthermore, often after the system is installed and the users have had a chance to work with it for sometimes, requirements that are not identified during requirement analysis phase will be uncovered. This occurs, since the experience with the software helps the user to define the needs more precisely. There might also be changes in the input data, the system environment and output formats. All these require modification of the software. The maintenance activities related to such modification fall under adaptive maintenance. The following features will be added in the near future to PlagiDetect:

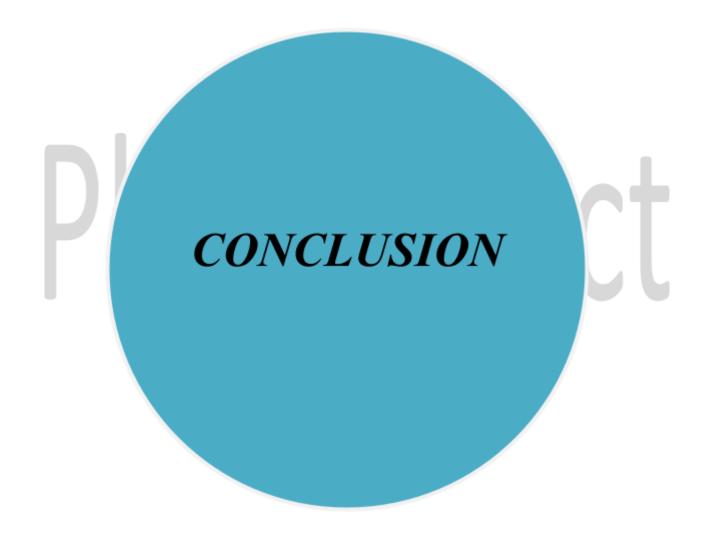
- Allowing other file formats like pdfs as well to get accessed by PlagiDetect
- Adding a fifth option of accessing the files from the local area network (LAN)
- Adding a sixth option of pasting or uploading a single text/text file to check plagiarism of that file with the entire internet.
- Hindi and Punjabi text will be allowed to be uploaded or accessed to successfully check plagiarism among these files.



CONTRIBUTION

A web application like PlagiDetect is need of the hour. It will contribute to the professional organizations like educational institutes in checking assignments. Also, it will assist researchers in getting the plagiarism percentage in their work. It will help eradicate academic dishonesty and contract cheating. The possibility of plagiarism can arise whether a single person performs it or multiple students contact the same person to complete their assignments. Whether knowingly or unknowingly, a piece of text can be prone to plagiarism. It is always significant as well as safe to apply plagiarism detection on a particular piece of writing. : A plagiarism detector helps you remove badly rephrased phrases by matching it in online repositories. You can have a printed copy of the plagiarism check that you can provide as evidence that you've taken the necessary precautions for any reason you are still questioned by your tutor on certain aspects of your paper.

The present features and the future updations are strong enough to have a vital contribution of the django web application – PlagiDetect to the society. Apart from the mentioned features and updations. PlagiDetect also focuses on developing a database of its own. This database can then be used to perform plagiarism among all the stored files (.txt, .pdf, and other supported formats). We aim to upgrade PlagiDetect more and more in the near future, so that it becomes extremely stronger option for people out there who perform plagiarism detection on daily basis. These days the plagiarism problem is skyrocketing these days and the craze of plagiarism is increasing day by day. There are some facts that need to understand about the importance of plagiarism checker. When anyone copies the content and use that content in their website, It results into lower page rank.



CONCLUSION

Web application like PlagiDetect helps people to detect plagiarism with flexible options to access text files. The present features and the future updations are strong enough to have a vital contribution of the django web application – PlagiDetect to the society. The SDLC (Software Development Life Cycle) has helped this project to move forward consistently and persistently. The present working four options are like a seed that will grow in the future to sow bigger fruits. The aim is to accomplish a successful and ever improving web application that can contribute to the society in much greater ways.

Concludingly, we would like to be heartily grateful to all our teachers, specially our Project guide for encouraging and motivating us at every step of our project work to keep moving forward with focus and concentration. PlagiDetect is just the starting of bigger, valuable, and meaningful projects to the society. Not only the features mentioned in this report, but also many more will be added as per the requirements of the future generation from time to time. It is a great opportunity for us to take this great project to much greater heights with new features and hence utilizing the maintenance phase with the proper implementation of the Agile Methodology. At the end, we would just like to conclude with determination in our hearts and minds to keep improving personally and professionally in our lives. In addition to this, also improving and enhancing PlagiDetect Django Web Application to greater altitudes.