Safe Storage: One Step Application for Storing all Private Files

BY

Saif Al Mahmud Mozumder
ID: 191-15-2517
AND

Merazul Islam Meraz
ID: 191-15-2385

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Professor Dr. Md. Ismail Jabiullah
Professor
Department of CSE
Daffodil International University

Co-Supervised By

Taslima Ferdaus Shuva
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2023

APPROVAL

This Project/internship titled **"Safe Storage: One Step Application for Storing all Private Files"**, submitted Saif Al Mahmud Mozumder, ID No:191-15-2517 and Merazul Islam Meraz, ID No:191-15-2385 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 25 January 2023.

BOARD OF EXAMINERS

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Professor Dr. Md. Ismail Jabiullah ,Professor ,Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

[illegible]

____

Name : Professor Dr. Md. Ismail Jabiullah
Professor
Department of CSE
Daffodil International University

Co-Supervised by:

[illegible] 22.01.23

____

Name : Taslima Ferdaus Shuva

Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

[illegible] 24-01-2023

____

Saif AL Mahmud Mozumder
ID: 191-15-2517
Department of CSE
Daffodil International University

Meraz

____

Merazul Islam Meraz
ID: 191-15-2385
Department of CSE

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Professor Dr. Md. Ismail Jabiullah**, Professor, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “Safe Storage: One Step Application for Storing all Private Files” to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Professor Dr. Touhid Bhuiyan**, Professor, and Head**,** Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.
Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

We use drive to store our important documents, records, images as well as memos. So it is a repository where the user can store, view, edit as well as download their own files as and when required. But the major disadvantage of these drives is that there is always a sole administrator which has access to all the accounts. So in case of any suspicious or malicious activity, he/she(admin) can have a look at our account. This enhanced us to engender an idea to implement a secure drive which will value the highly rated motto of furtiveness as well as privacy which is the most desired principle in today's era. Overcome the problem of Administrator's access to the contents of Database even though the contents are restricted to the authenticated user. Hence to overcome this issue the files will be stored in an encrypted format and the contents will not be visible to the Administrator. Also to keep the files secured from unauthorized users or any malicious activity, the Drive will be immune to attacks and penetrations using encryption and two-step authentication.

TABLE OF CONTENTS

CONTENTS	
Board of examiners	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
CHAPTER	
CHAPTER 1: INTRODUCTION	**1-3**
1.1 introduction	1
1.2 Problem Definition	1
1.3 Scope	1-2
1.4 Hardware and Software requirements	2-3
CHAPTER 2 : LITERATURE SURVEY	**3-4**
2.1 Introduction	3
2.2 Summery	4
CHAPTER 3 : PROJECT DESIGN	**3**
3.1 Proposed System model	3-9
3.2 Software Project management Plan	9-12
CHAPTER 4: Implementation and experimentation	**12-28**
4.1 Proposed system model implementation	12-17
4.2 additional details	17

4.3 Software Testing	17-24
4.4 Experimental results and its analysis	24-28
CHAPTER 5 : Comparative studies	**31**
5.1 Comparison	31
CHAPTER 6: Conclusion	**31-32**
6.1 Summery	31-32
6.2 Future work	32
References	33

LIST OF FIGURES

FIGURE	PAGE NO
class diagram	8
System Architecture	9
Face Recognition Flow chart	15
Work Flow of CBC	16
Register flow chart	20
Login Page	26
Register page	27
Face Detection	28
Landing Page	29

LIST OF TABLES

TABLES	**PAGE NO**
Timeline Chart	10
Activity table	11-12
Module Description	13
Testing chart	18
Test module	18-19
Test Cases	22-23

Chapter 1: Introduction

1.1 Introduction

The fact that everyone can grasp the system is certainly file-based storage's largest benefit .A hierarchical structure is not just recognizable from IT, but it is also employed in theory in home file folders .Scaling a system with file storage is often also relatively simple .It is easy to add a new storage device to the network, such as a new NAS server, if more capacity is required .Additionally, the memory is accessible to and changeable by several network users simultaneously .Although navigating gets harder as files grow in size, file-level storage is theoretically highly scalable and affordable.

As a result, opening individual files takes longer .Users' files must be protected so that, even if an attacker or hacker gains access to the server, they cannot access the actual files .Another assault that is avoidable is the guy in the middle one .The encryption and decryption techniques should prevent the server from having all the necessary information to decrypt any encrypted files kept on the server.

1.2 Problem Definition

Solve the issue of the administrator having access to the database's contents despite the contents only being available to authorized users. In order to solve this problem, files will be kept in an encrypted manner with their contents hidden from the Administrator. Encryption and two-step authentication will make the Drive resistant to assaults and penetrations, further securing the contents from unauthorized users or any unwanted activities.

1.3 Scope

Providing Authentication and Encryption.

Connectivity between the contents of the database and the hosting server of the database for successful transactions.

Maintain the system and fix the bugs and upgrade upto ONE month after commissioning.

Procurement of hardware for using the system is OUT OF SCOPE of this project.

1.4 Hardware and Software Requirements

The software will operate on mobile phones, tablets, laptops etc. In the existing scenario, a website will be created to deploy this software.

In the future, the possibility of creating an iOS app will be taken into consideration.

Hardware Requirements:

Quad Core 1.2GHz

RAM (GB) 2 GB

Internal (GB) 16 GB

Operating System:

Windows 7 and advanced

Requirements gathering:

Functional Requirements

1. The system shall allow a user to Store all documents/files.
2. The system shall allow a user to view his/her Stored documents
3. The system shall allow a user to update his Picture for recognition.
4. The system shall store the files in a database in encrypted format.
5. The system shall display all the stored files of the user.
6. The system shall accept an image of the user only from the webcam to assure the presence of the user.
7. The system shall upload files less than 100mb at initial stage.

Non-Functional Requirements

1. Frames rate

The minimum frame rate must be twenty frames per second. The average frame rate must be greater than 30. Frame rate can be monitored directly from the graphic engine.

2. Response time

The average response time between click and reaction must be less than 0.5 seconds. The maximum response time between click and reaction must be two seconds. Adding some simple classes and methods that will compute and display the time needed to process any operation can test these requirements.

3. Required resources

The file can be any size ranging from few MB to GB. The Software must use less than 50MB of disk space.

4. Platform

The project must run on all servers.

5. Maintainability

The code written for the software must be maintainable. Also adding documentation will improve the maintainability scale of the system.

Chapter 2: Literature Survey

2.1 Introduction

In order to safeguard the confidentiality, integrity, and accessibility of files saved on a computer or network, cryptographic file storage is used. You may begin a literature study by outlining the value of protecting stored files and the many dangers they might encounter .Then, you might go over the several cryptographic methods that have been suggested or used for file storage, such as symmetric and asymmetric encryption, hashing, digital signatures, and others. The management of keys and other cryptographic parameters is a crucial component of cryptographic file storage. Aspects like key creation, distribution, storage, and revocation might fall under this category. You might also talk about the compromises made by various key management strategies, such as the choice between longer and shorter keys in terms of convenience vs. security.

2.2 Summary

Wherever we upload files on the internet however secure it might be the database admin always has access to them. Even the accounts maintained by our organizations can look into the contents of the file, So to overcome this we have designed an application that stores our sensitive files in encrypted form on the database with our private key We have used the fernet module to encrypt files that use AES in CBC mode with a 128 bit key as mentioned below. Using this application we can store any type of file i.e. mp4,wav,mp3,txt,xls,pdf,png, etc. Also accessing these files is super easy as we do not have to remember anything except our set password that can be alphanumeric as short as 4 characters long and the rest is handled by our application.

Chapter 3: Project Design

3.1 Proposed System Model

Here, the components, architecture, modules, interfaces, and data for a system are defined in accordance with the given specifications .A safe end-to-end encrypted storage application is Secure Drive.

The application's several components include the homepage of our website, a drive page where users can save and view their files, and a profile page where users can see how many files they have stored and how much storage they have consumed.

1.Authentication: The User must sign up using both a face ID and an email address in order to be authenticated into the drive.

2.Drive Page: For storing computer files. Once the file has been decrypted with the crypto key, it may be viewed of the specified user. The files are stored in an orderly manner with time of upload and the size of the file.
3. The profile page consists of the details of the user such as number of files stored and storage used also the username and name of the user
1. Constraints :The system must be compatible with a range of operating systems and bias. The system must be biddable with applicable data protection regulations and norms.

2. hypotheticals and Dependences :It's assumed that druggies have the necessary tackle and software to pierce and use the system. The system is dependent on a secure garçon for storing translated lines.

3. stoner Classes and Characteristics: The system will be used by individualities and associations with a need to securely store sensitive data. druggies may have a range of specialized moxie, from neophyte to advanced.

4. stoner terrain: The system will be penetrated through a web cyber surfer on a variety of bias, including desktop computers, laptops, and mobile bias.

5. Design and perpetration Constraints: The system must be designed and enforced in a way that ensures security and confidentiality of translated lines and keys. The system must be scalable and suitable to handle a large number of druggies and lines. The system must be stoner-friendly and easy to use.

6. hypotheticals and Dependences :It's assumed that druggies have the necessary tackle and software to pierce and use the system. The system is dependent on a secure garçon for storing translated lines.

7. supplements :A list of cryptographic algorithms and crucial operation styles that the system will support. A list of data protection regulations and norms that the system must misbehave with.

System Features

Once a user Registers & creates an account , they can store their documents on the server without worrying about their privacy.
Admin may help users to reset the password but cannot view into contents of the database as they are all encrypted using a private key.
Whenever the authorized user wants to view their own stored content it is decrypted and viewed/downloaded.

Operating Environment

The operating environment for Secure Drive is listed as follows:
Linux Hosting is by default provided
Browser preferably Chrome or Firefox
Client/Server architecture
Database: SQLite3
Platform: Python (Django)

Design and Implementation Constraints

Python : Frontend
OpenCV Framework
API : For compiling & running codes.
IDE will limit the languages used during submissions for example C,C++,Java ,Python.
Admin will be responsible for maintaining the contests.
Proper password protection by encryption is needed, else security issues might happen.
Smooth and proper working of a query system.
Communication between the pages and the database in a seamless fashion

External Interface Requirements

User Interfaces

Login/Register page will be provided.
A Profile section for users to modify their personal details.
Once the contest is created ,it will be displayed on the screen with the time remaining for start.

Communications Interfaces

The backend will involve a REST API, which will communicate with the client over HTTP/ HTTPS protocol.
This data will be exchanged via JSON objects.
Mail server to send emails over SMTP protocol.
JSON Web Tokens (JWT) will be stored in the client’s browser so as to perform authentication. The above emails will contain JWTs which will help assign people to certain tasks.
The classified information within each JSON object received at the backend will be stored in the DB in an encrypted format.
Rate limit for the number of requests sent to the API will be implemented.

Class Diagram:

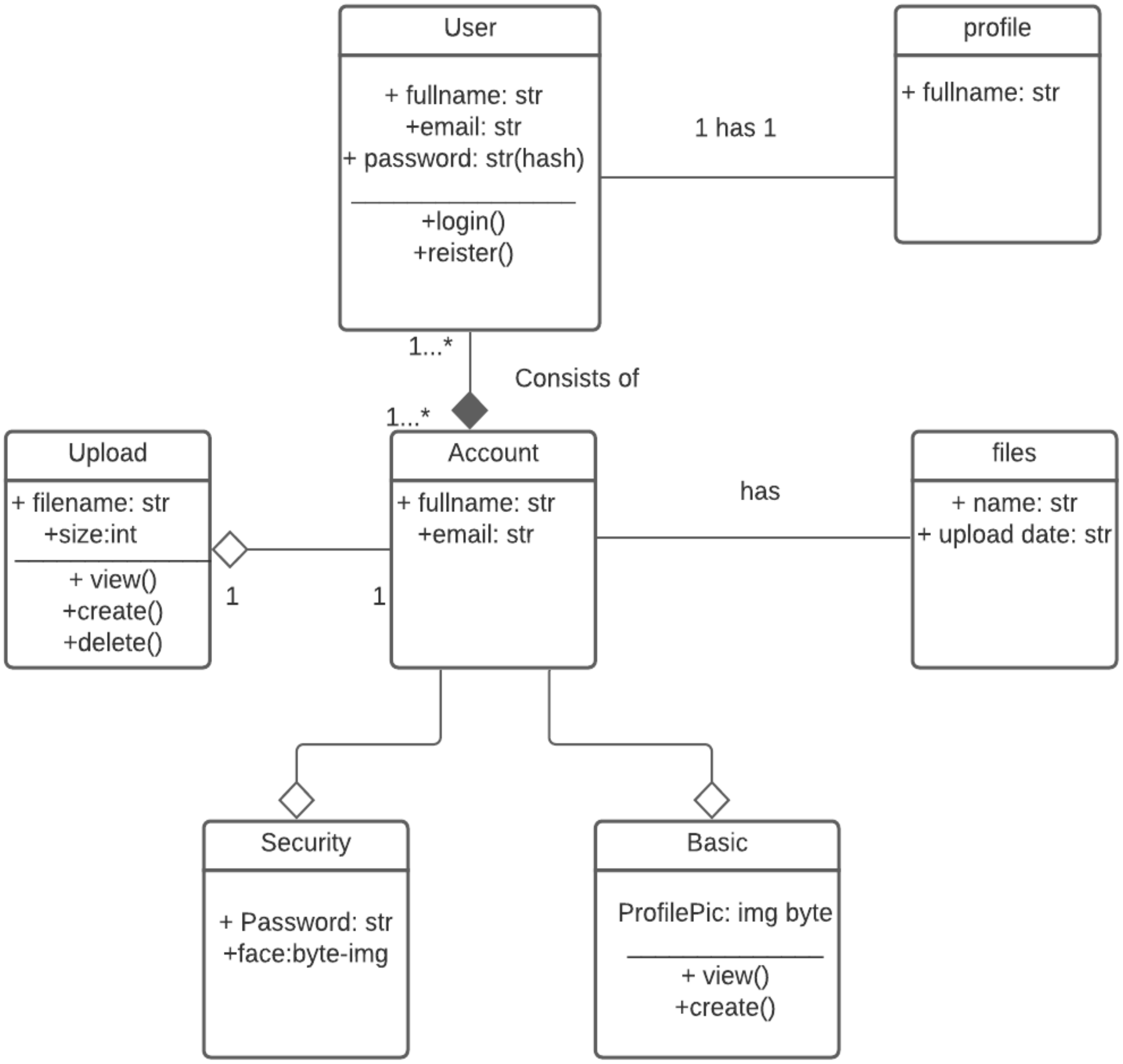


Figure3.1 : class diagram

System Architecture:

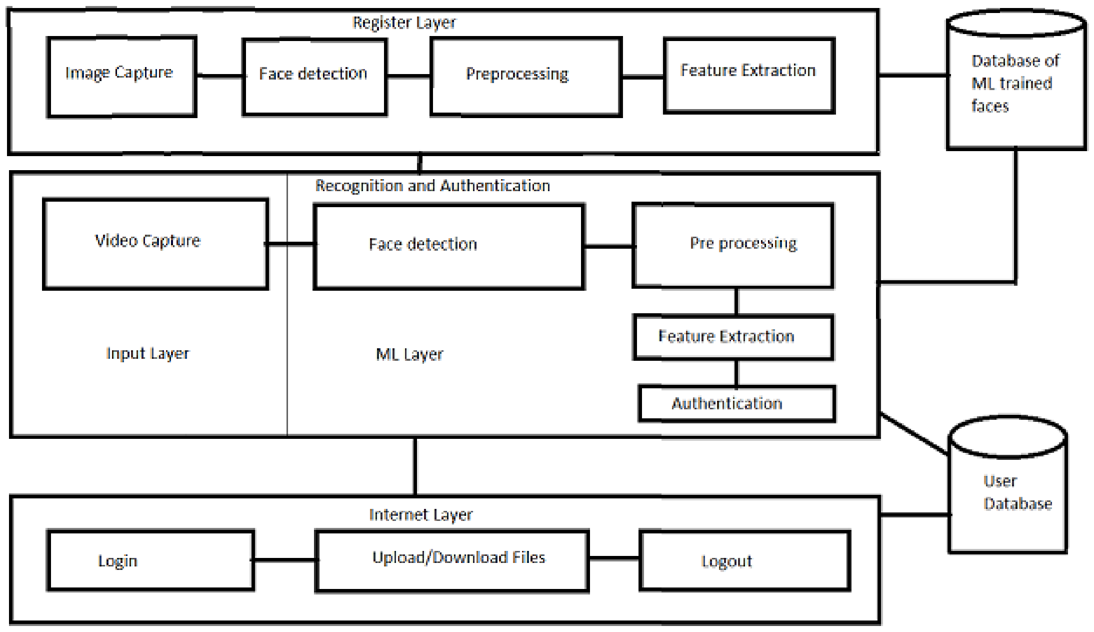


Figure3.2: System Architecture

The system uses Python on the backend and frontend and mongoDB as well as SQLite as the database.

Main Interfaces:

1. Authentication (Register and Login)
2. Profile page
3. Drive page
4. Landing page

3.2 Software Project Management Plan

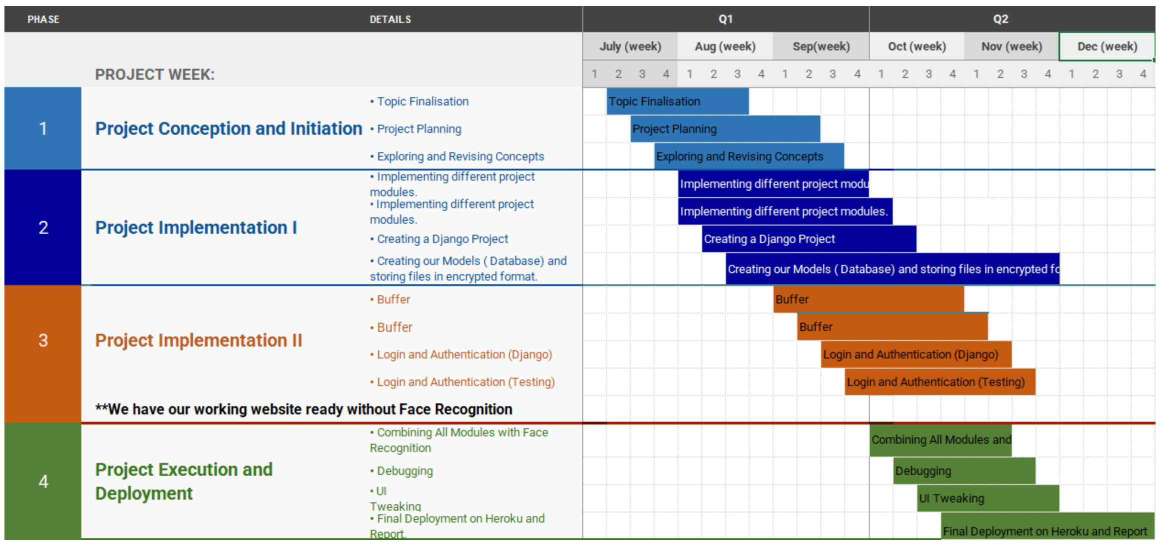
The main functionality of the app can be divided into 5 different parts.

1. Authentication (Register and Login)
2. Uploading of files
3. Storage of files in an encrypted format

Timeline:

The project was built in the span of 4 months. The project timeline is shown below:

Table 3.1: Timeline Chart



PHASE	DETAILS	
	PROJECT WEEK:	
1	Project Conception and Initiation	• Topic Finalisation • Project Planning • Exploring and Revising Concepts
2	Project Implementation I	• Implementing different project modules. • Implementing different project modules. • Creating a Django Project • Creating our Models (Database) and storing files in encrypted format.
3	Project Implementation II	• Buffer • Buffer • Login and Authentication (Django) • Login and Authentication (Testing)
	**We have our working website ready without Face Recognition	
4	Project Execution and Deployment	• Combining All Modules with Face Recognition • Debugging • UI Tweaking • Final Deployment on Heroku and Report.

Q1
Q2
July (week)
Aug (week)
Sep(week)
Oct (week)
Nov (week)
Dec (week)
1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
Topic Finalisation
Project Planning
Exploring and Revising Concepts
Implementing different project modu
Implementing different project modules.
Creating a Django Project
Creating our Models (Database) and storing files in encrypted fo
Buffer
Buffer
Login and Authentication (Django)
Login and Authentication (Testing)
Combining All Modules and
Debugging
UI Tweaking
Final Deployment on Heroku and Report

Responsibility Matrix:

On the technological side, we intend to evenly distribute the effort. We have split the sections in order to look for creative designs, test them, put them into practice, and comprehend the process. Job sharing is carried out in accordance with our areas of interest.

The team's fundamental workload is organized as follows.

Saif AL Mahmud: UI/UX Implementation, Testing. Cryptography algorithms, Face recognition module

Merazul Islam Meraz: Testing

Table3.2 : Activity table

Activity	Saif Al Mahmud	Merazul Islam Meraz
1.Requirement Gathering		
1.1 Interaction with customer	Creator, Executor, Approver	Reviewer
1.2 Preparing SRS	Creator, Executor, Approver	Reviewer
2. Design		
2.1 Preparing Block diagram	Creator, Executor, Approver	Reviewer
2.2 Writing Functional Requirements	Creator, Executor, Approver	Reviewer
2.3 Writing Non-Functional Requirements	Creator, Executor, Approver	Reviewer
2.4 Developing Use Case	Creator, Executor, Approver	Reviewer
2.5 Developing Test Cases	Creator, Executor, Approver	Reviewer
4. **Coding**		
4.1 Unit 1	Creator, Executor, Approver	Reviewer
4.2 Unit 2	Creator, Executor, Approver	Reviewer
4.3 Front end/ UI	Creator, Executor, Approver	Reviewer

5. **Testing**		
5.1 Frontend	Creator, Executor, Approver	Reviewer
5.2 Cryptography algorithms	Creator, Executor, Approver	Reviewer
5.3 Database Management	Creator, Executor, Approver	Reviewer

Chapter 4: Implementation

4.1 Proposed System Model Implementation

Methods:

In order to develop a convenient and secure storage solution, we followed the following steps:

1. Researching and selecting an appropriate encryption algorithm: We conducted extensive research on various encryption algorithms to determine which one would be the most suitable for our storage solution. We considered factors such as security, efficiency, and ease of implementation. Ultimately, we selected the AES (Advanced Encryption Standard) algorithm, which is widely recognized as one of the most secure and efficient encryption methods available.
2. Developing a user interface: We designed a user interface that would allow users to easily upload, download, and manage their files. This included features such as a file explorer and options for organizing and categorizing files. We also incorporated security measures into the interface, including password protection and the ability to set permissions for individual files.
3. Implementing the encryption and decryption process: We wrote code to handle the encryption and decryption of files as they are uploaded and downloaded by the user. This involved integrating the AES algorithm into the system and implementing the necessary logic to handle the encryption and decryption process.
4. Testing and debugging: Before making the storage solution available to users, we conducted thorough testing to ensure that it was functioning properly and that all security measures were in place. Any issues that were discovered during testing were addressed through debugging and code changes.

Modules Description:

Table3.3 : Module Description

Module	Name, Definition, purpose
User Authentication	*To handle login and signup using email and face recognition*
Profile	*To get the number of files of a particular user and storage occupied by those files*
Storage module	*To store and retrieve files securely through encryption and decryption*

4.2 Algorithms

A file's contents will be kept secret from the administrator and maintained encrypted. The Drive will be resistant to attacks and penetrations thanks to encryption and two-step authentication, thus protecting the data from unauthorized users or other undesirable actions. Users can connect into a device or system by presenting their face to a camera with a face detection login system, which employs facial recognition technology. Traditional login techniques like entering a password or using a fingerprint scanner can be replaced with this kind of system.

face-recognition-models:

Identifying and confirming a person's identity from an image or video is a computer vision challenge known as face recognition.

Due to its vast libraries and frameworks for machine learning and computer vision, Python is a well-liked programming language for implementing these models. Face recognition may be performed using a variety of techniques and models.

You must perform the following actions in order to develop a face detection login system:

Install a camera that can provide detailed pictures of the user's face.

Utilize software to identify and separate the face from the camera-captured photos.

In order to assess whether the user is recognized, the system will compare the extracted face to a database of recognized faces.

The system will let the user log in if they are identified.

If the user cannot be identified, the system could request other forms of identification, such a password or a fingerprint scan.

The system may also employ extra features like liveness detection, which verifies that the face being displayed to the camera is a live face and not a picture or video, to increase security.

Numerous methods may be used to do this, including template matching and machine learning algorithms to compare the face to a known template and distinguish facial traits.

The dlib package, which includes a variety of methods for recognizing and tracking faces in photos and videos, is one well-liked library for face recognition in Python. A pre-trained facial recognition model is available in the library that may be used to find and contrast faces in pictures.

OpenCV (Open Computer Vision), an open-source library for computer vision applications, is another well-liked library for face recognition in Python.

The Viola-Jones algorithm, a popular technique for identifying faces in pictures, is one of many face detection and analysis methods included in OpenCV.

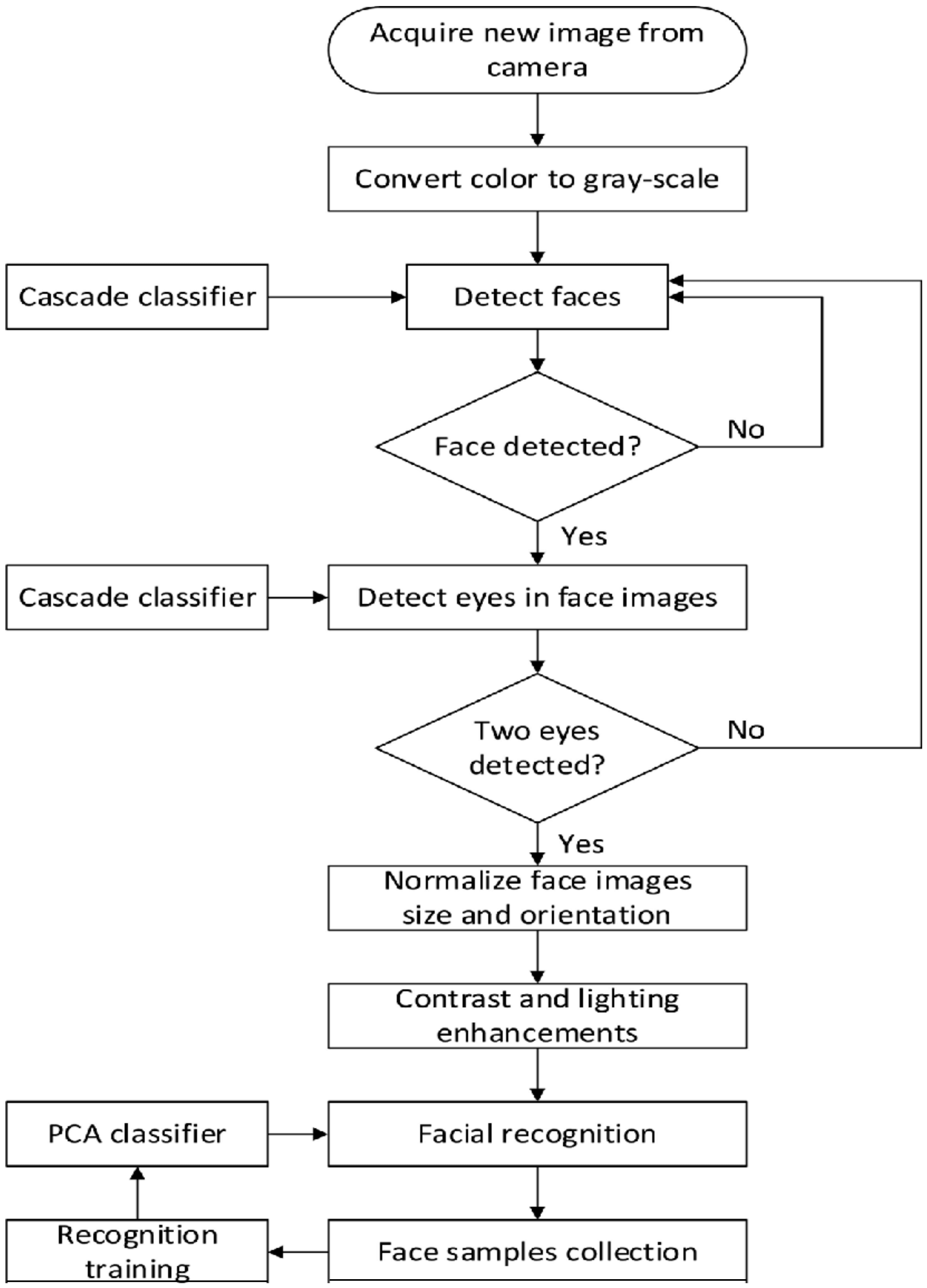


Figure 4.1: Face Recognition Flow chart

Password hash :

The password hash algorithm is a mathematical operation that converts a password into a fixed-length character string known as a hash. Since the mathematical process used to build the hash is one-way, it is computationally impossible to reconstruct the original password from the hash. One of the key advantages of utilizing a password hash method is that it enables safe password storage without requiring the storage of the real password. This is significant since it ensures that the original passwords cannot be retrieved, even if an attacker manages to access the password database.

With a 128-bit hash value, the Message-Digest Algorithm 5 (MD5) is a popular cryptographic hash function. Ronald Rivest created it in 1991 as a means of producing a distinct, fixed-size representation of a message or data file. When using the MD5 hash algorithm, an input message is divided into blocks of a specific size. After that, each block is subjected to a series of mathematical procedures to produce a fixed-size hash value. The resultant hash value is commonly expressed as a 32-character hexadecimal string and is a string of 128 bits (16 bytes) in length. The integrity of data files may be checked, unique keys can be created to identify items in a database, and hash-based message authentication codes can be created using the popular algorithm MD5 (HMACs).

It should be emphasized, nevertheless, that MD5 is regarded as being quite weak in terms of cryptography and is susceptible to specific kinds of attacks.

It is no longer advised to utilize SHA-1 in new systems and should be changed with SHA-2 or SHA-3, which are more secure hash functions.

Encryption And Decryption:

Block encryption algorithms like AES use CBC (Cipher Block Chaining) as its method of operation (Advanced Encryption Standard). It entails breaking up the plaintext message into fixed-size blocks and employing the cipher and an initialization vector to encrypt each block (IV). Then, using the IV—a random value used to make sure that the same plaintext message does not yield the same ciphertext when encrypted several times—the resultant ciphertext blocks are chained together.

The fact that CBC mode resists attacks that entail the modification of certain ciphertext blocks is one of its benefits. This is so that tampering can be easily detected since any modifications to the ciphertext would cause the succeeding blocks to be erroneously decrypted.

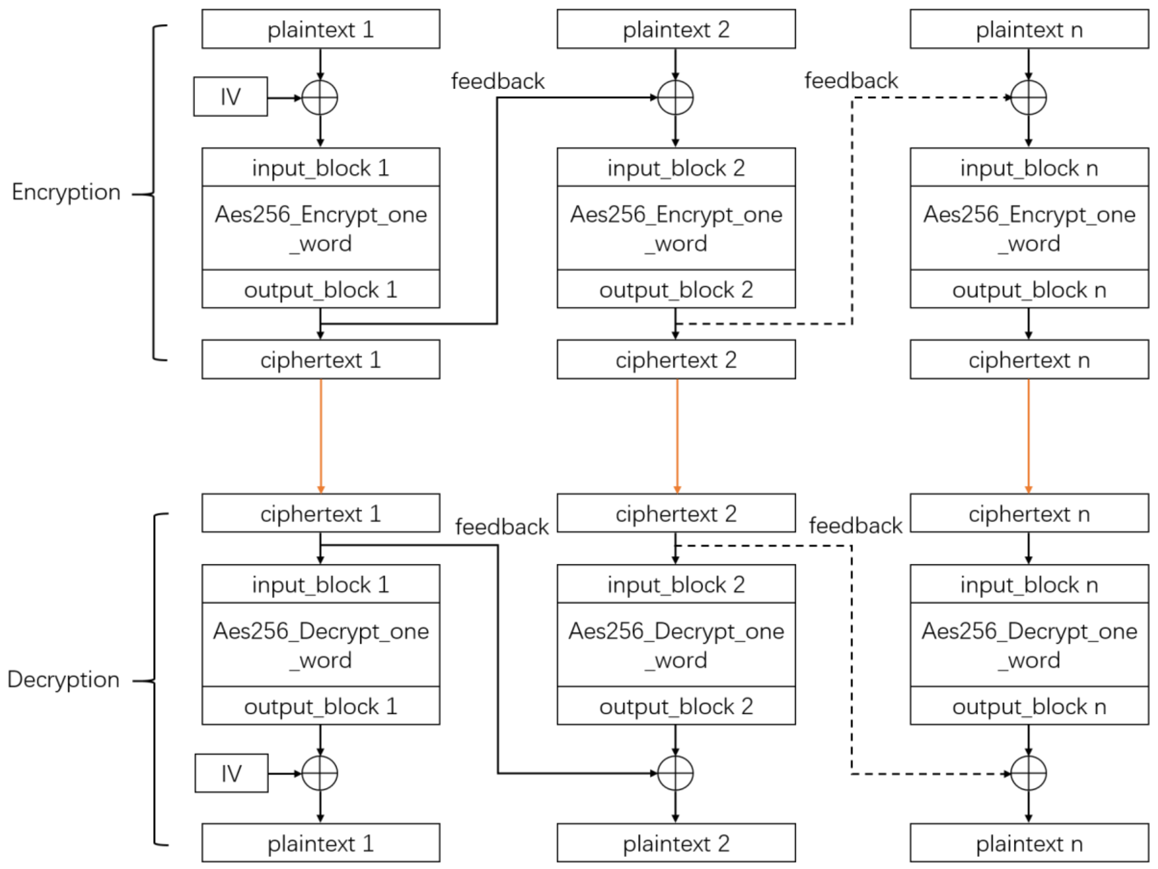


Figure 4.2: Work Flow of CBC

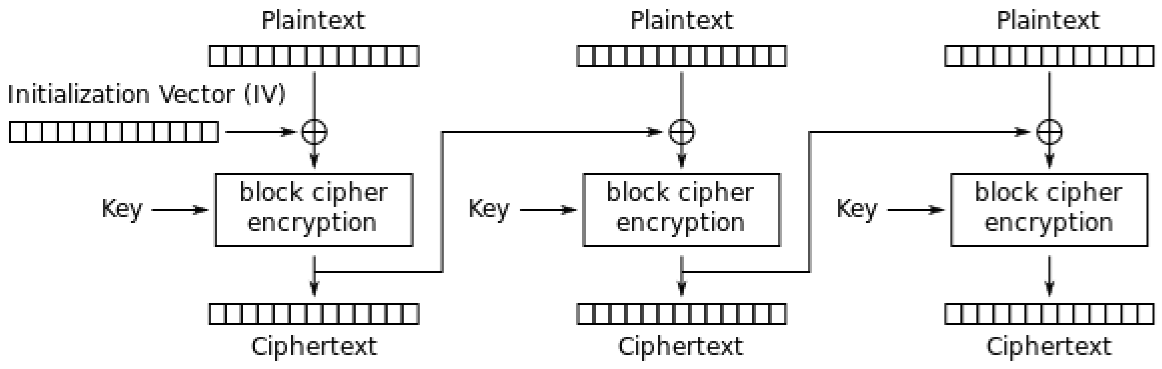


Figure 4.3 :CBC

4.2 Additional Details

We were suggested by our project guide to implement a feature which is not present in WhatsApp or telegram

This suggestion was taken into account and implemented. This resulted in the application having one extra unique feature

The feature that we decided to add was scheduling messages which is a feature that is not present in either telegram or WhatsApp .

4.3 Software Testing

Following the integration of the subsystems and components listed in the Integration Build Plan for the Prototype, the architectural prototype will be subjected to system and integration testing.

Building the architectural prototype served the objective of evaluating the viability and efficiency of the chosen architecture.

At this early stage, it is crucial to verify the performance of the system as well as all system and subsystem interfaces.

The prototype won't be put through testing for features and functioning.

Test Schedule

The project was built in the span of 4 months. The project timeline is shown below:

Table 4.1: testing chart

Start Date	End Date	Task Name	Milestone
10-11-2022	11-11-2022	Authentication (Login/Sign Up)	Documentation and feedback.
23-11-2022	25-11-2022	File Encryption and Decryption	Developer's comments, automated tools report
01-12-2022	02-12-2022	File Uploading	User's feedback, system performance report
15-12-2022	16-12-2022	Face Recognition Module	User's feedback

Table 4.2: test module

Test Module	Test Case	Inputs	Expected Output	Actual Output
Authentication	The user should register with valid email and password	Valid account Invalid password	If the account exists, an indication of the same should be made. If not, the registration is successfully done. If the password created by the user does not follow the criteria, mention the same.	The user details are stored in the server on successful registration.
Encryption And Decryption	When the user stores his files in the drive they are stored in an encrypted manner which can only be decrypted using the user's crypto key. Then he can view the file.	Choosing file	Storing file in an encrypted manner	File stored successfully.

White Box Testing

1) **Cyclomatic Complex**
   Registration:

   Flowchart:

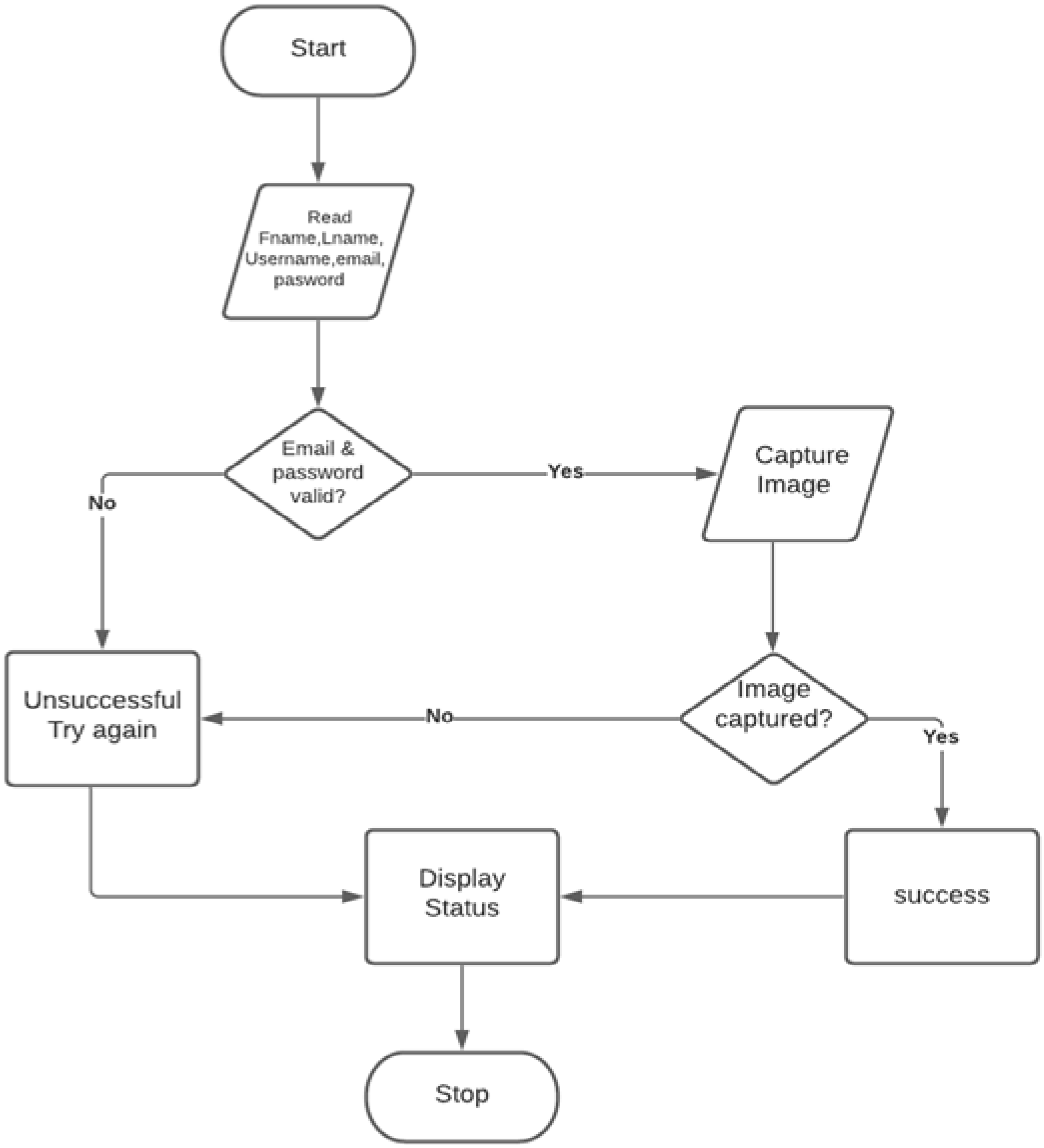


Figure 4.4: Register flow chart

Flow Graph

no. of edges=E=8

no. of nodes=N=7

no. of predicate nodes=P=2

Cyclomatic Complexity:

V(G) = Number of regions = 3

V(G)=E–N+2=8-7+2=3

V(G)=P+1=2+1=3

Therefore the cyclomatic complexity of the graph is 3

Generated Test cases:

Paths:

P1: 1,2,3,4,5,6,7,8,9,10

P2: 1,2,3,4,5,7,8,6,10

P3: 1,2,3,4,5,6,10

Test Case for P1:

Input valid Registration details

Face captured

Input correct password of specified length

Expected Output:

Registration Successful

Test Case for P2:

Input valid Registration details

Enter password other than specified length/Face not captured

Expected Output:

Registration Unsuccessful

<u>Login using Password:</u>

EC1 :- Correct username and password , correct face

EC2 :- Incorrect username or password , correct face

EC3 :- Incorrect username and password , correct face

EC4 :- Incorrect username (and/or) password , Incorrect face

Table4.3 : Test Cases

Test Case	Input Face	Expected Output	Equivalence class tested
TC1	Correct username, correct password, correct face	user is logged in>Profile page	EC1
TC2	Wrong username, correct password, correct face	Login page (No passthrough)	EC2
TC3	Wrong username, wrong Password, correct face	Login page (No passthrough)	EC3
TC4	Incorrect username, Incorrect password, Incorrect face	Login page (No passthrough)	EC4

BVA

Password length - minimum 4 characters and maximum 12 characters (any)

EC1 = {0,1,2,3,4,5}

EC2 = {6,7,8,9,10,11,12,13,14}

EC3 = {15,16,....}

If password length < 4 invalid length and raise an error.

If password length >4 and password length <12 accept the password and register user.

If password length > 12 invalid length and raise an error.

For Eg- test password is a valid password of length 12, *pas* is an invalid password of length of 3

4.4 Experimental Results and Analysis

Screenshots:

- **Login:**

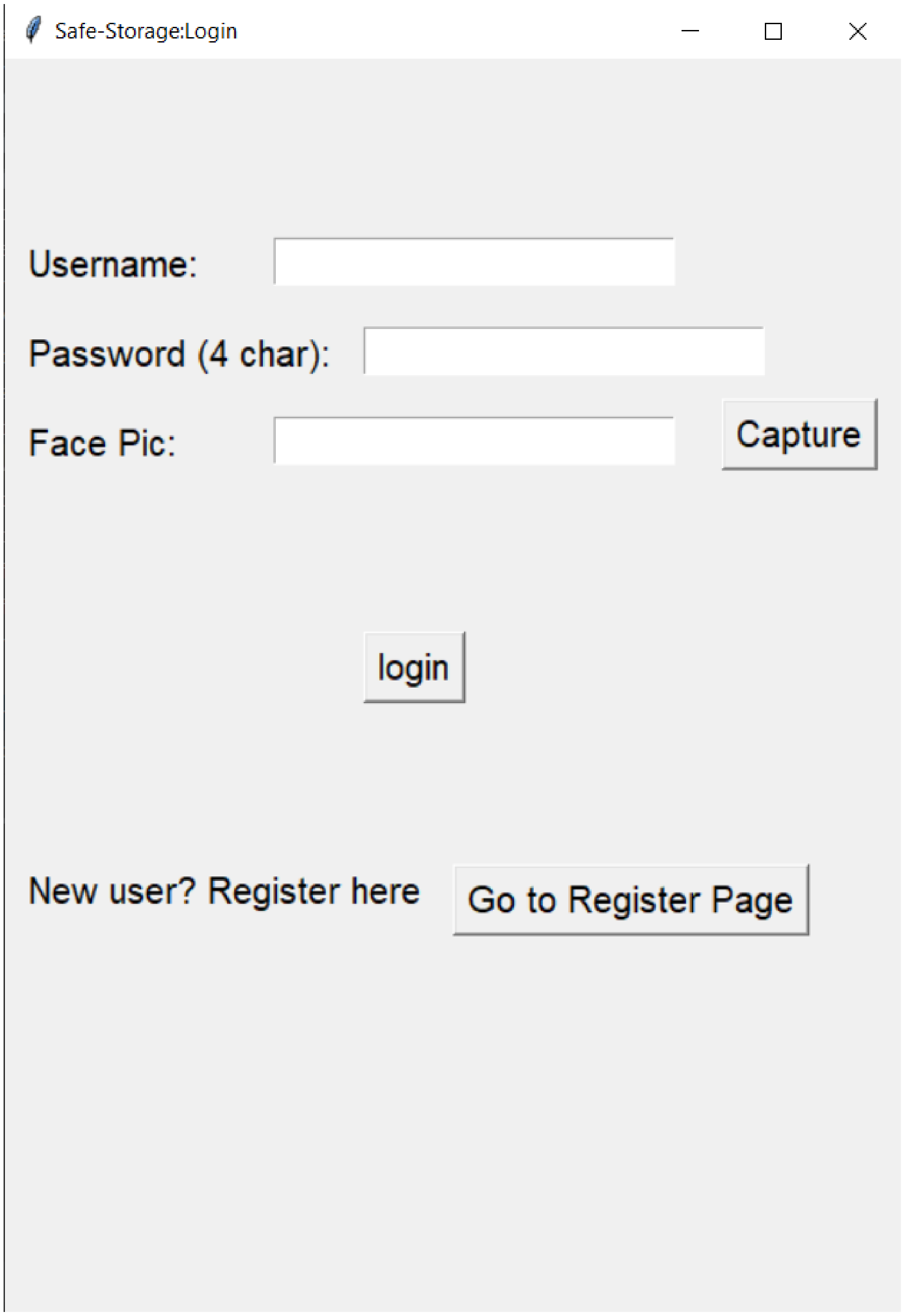


Figure 4.5 : Login Page

- **Register**

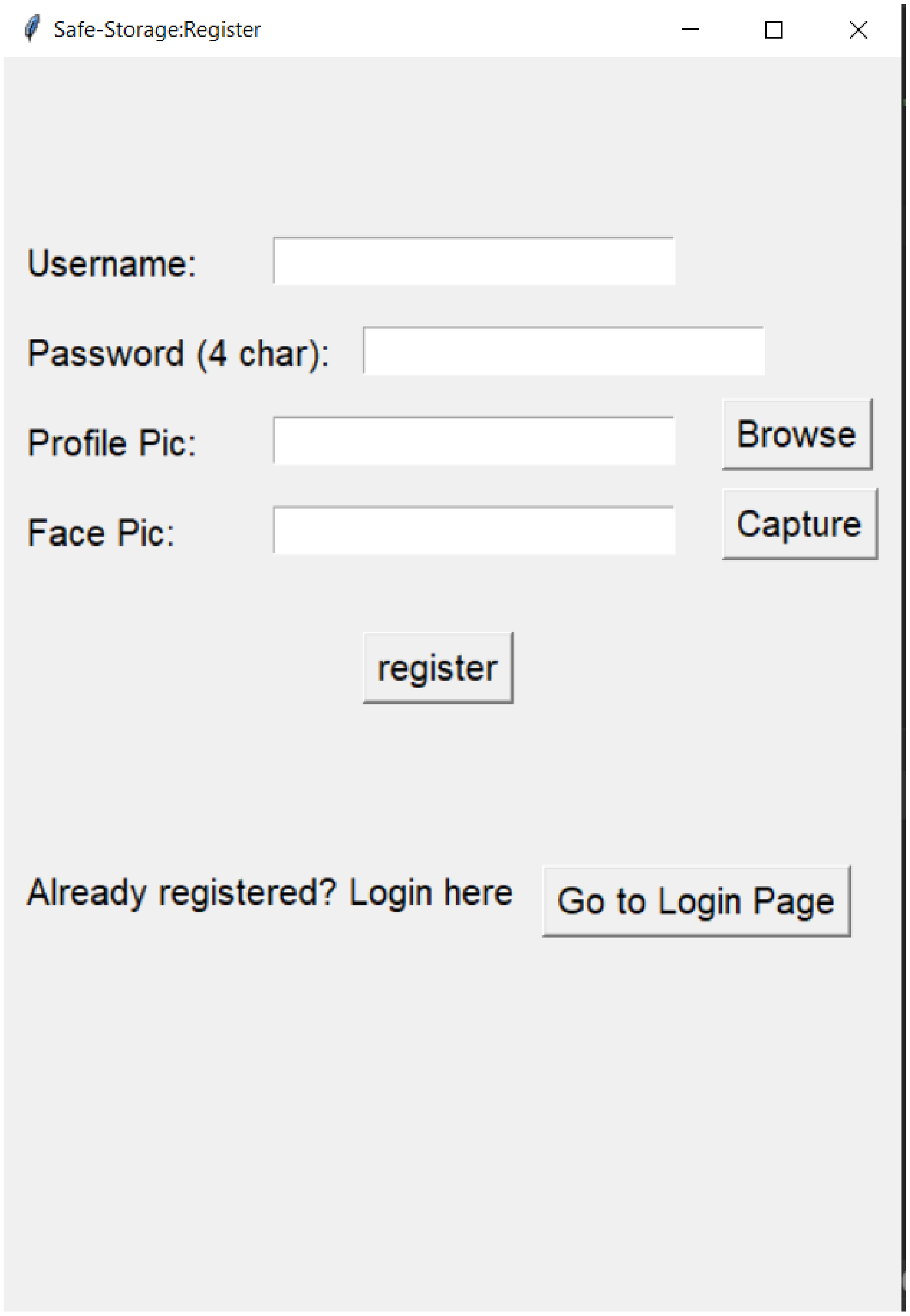
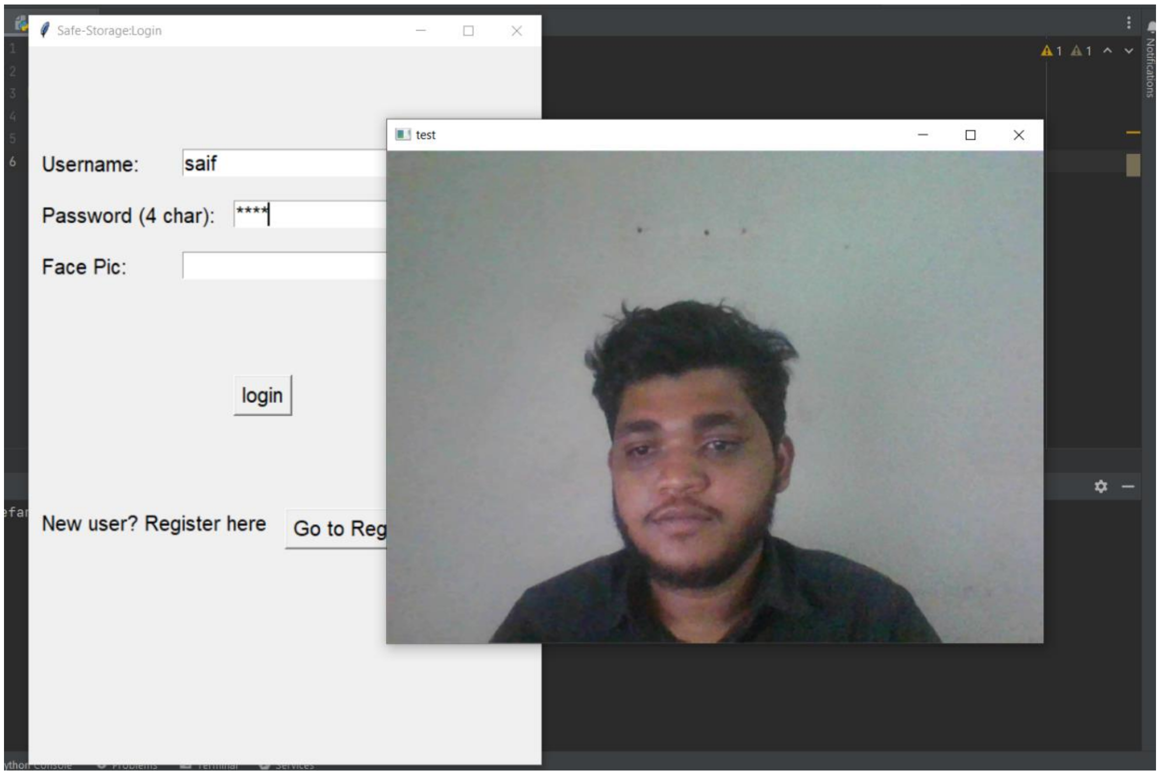


Figure 4.6 : register page

Face Recognition:

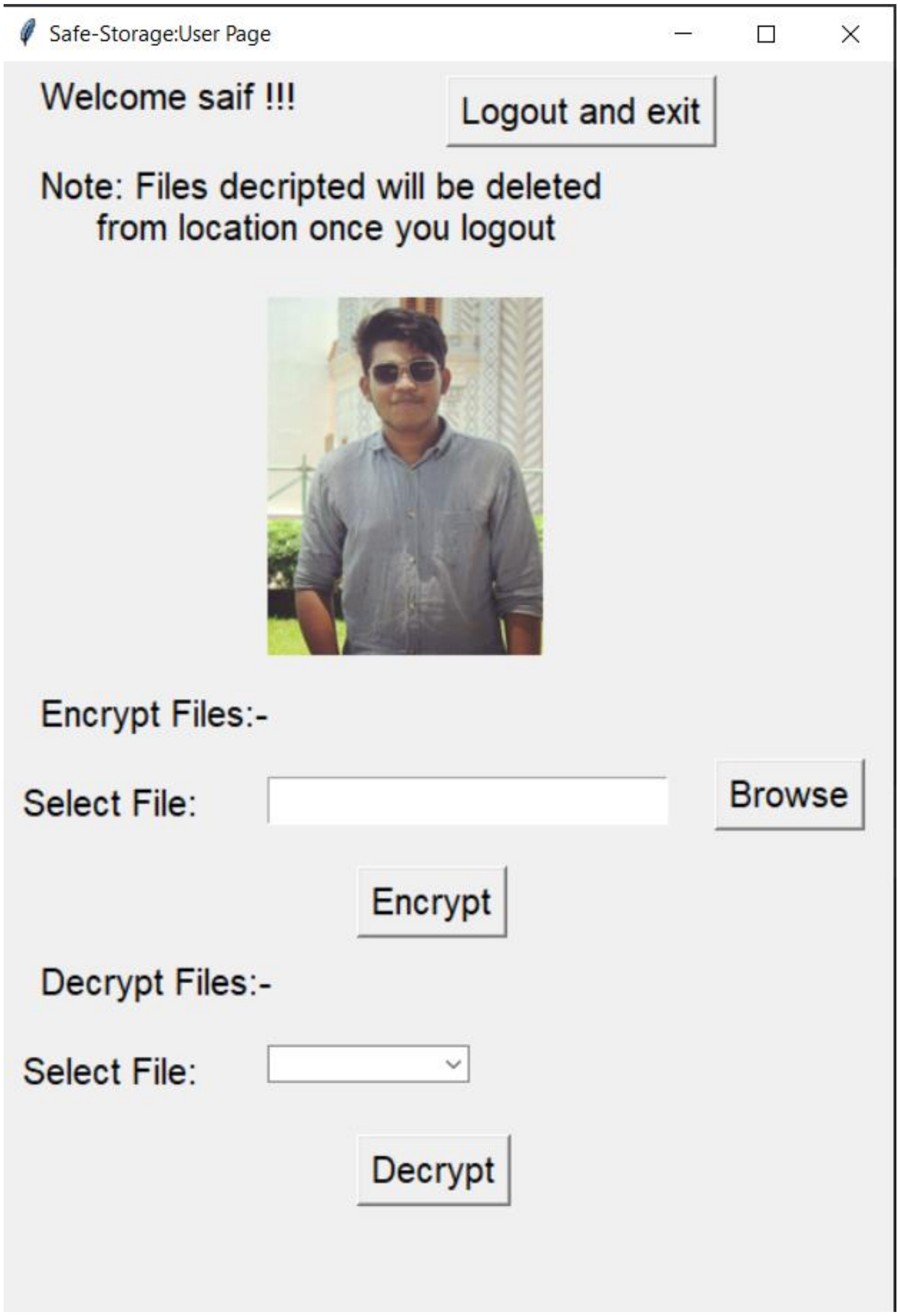


Username: saif

Password (4 char): ****

Figure 4.7 : face recognition

Landing Page:


Safe-Storage:User Page

Welcome saif !!!

Logout and exit

Note: Files decripted will be deleted
from location once you logout

Encrypt Files:-

Select File:

Browse

Encrypt

Decrypt Files:-

Select File:

Decrypt

Figure 4.8 : landing Page

Chapter 5: Comparative studies

5.1 Comparison

Systems for storing files using encryption are intended to safeguard the availability, confidentiality, and integrity of data. These systems safeguard data in storage, whether it is on a local device or in the cloud, using a variety of encryption techniques and protocols. Systems used to store cryptographic files frequently include

EFS -Users can encrypt certain files and folders on their local computer using the Encrypting File System (EFS) feature of the Windows operating system. A public and private key pair is used in public key cryptography, which is what EFS does. The data is encrypted using the public key, and then decrypted using the private key.

BitLocker - The Windows operating system also includes the encryption function BitLocker. Users may use it to encrypt any section of their hard disk, including the boot and system partitions. BitLocker supports a number of authentication techniques, including passwords and biometric verification, and employs AES encryption.

TrueCrypt-Create encrypted volumes on a local device or in the cloud with TrueCrypt, a free and open-source encryption tool. It works with a number of encryption techniques and is compatible with Linux, macOS, and Windows platforms.

VeraCrypt - VeraCrypt is an additional free, open-source encryption program built on the TrueCrypt platform. It enhances and adds new capabilities including the ability to encrypt boot loaders and system partitions as well as support for bigger encryption keys.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

We thus implemented our safe, a place to store files in an encrypted format so that even the admin could not view those files. Our project mainly focuses on privacy and security of files of the users, we also provided a simple and fast UI. The creation of a practical and safe storage option is a crucial first step in preventing unwanted access to sensitive data .We have developed a method that makes it possible for users to keep their files securely since it uses encryption to guard against online dangers .By offering a safe and practical location to keep their information, this solution has the potential to help a wide spectrum of people and organizations .The use of encryption can provide consumers other advantages beyond those related to security. We aim to continue developing our application in the future.

6.2 Scope of Future Work

The future scope of this project includes the functionality to upload folders, edit the names of files as well as folders, sharing with others, offline viewing, ability to search desired files and create groups. As the application progresses we also plan on making the application even more secure.

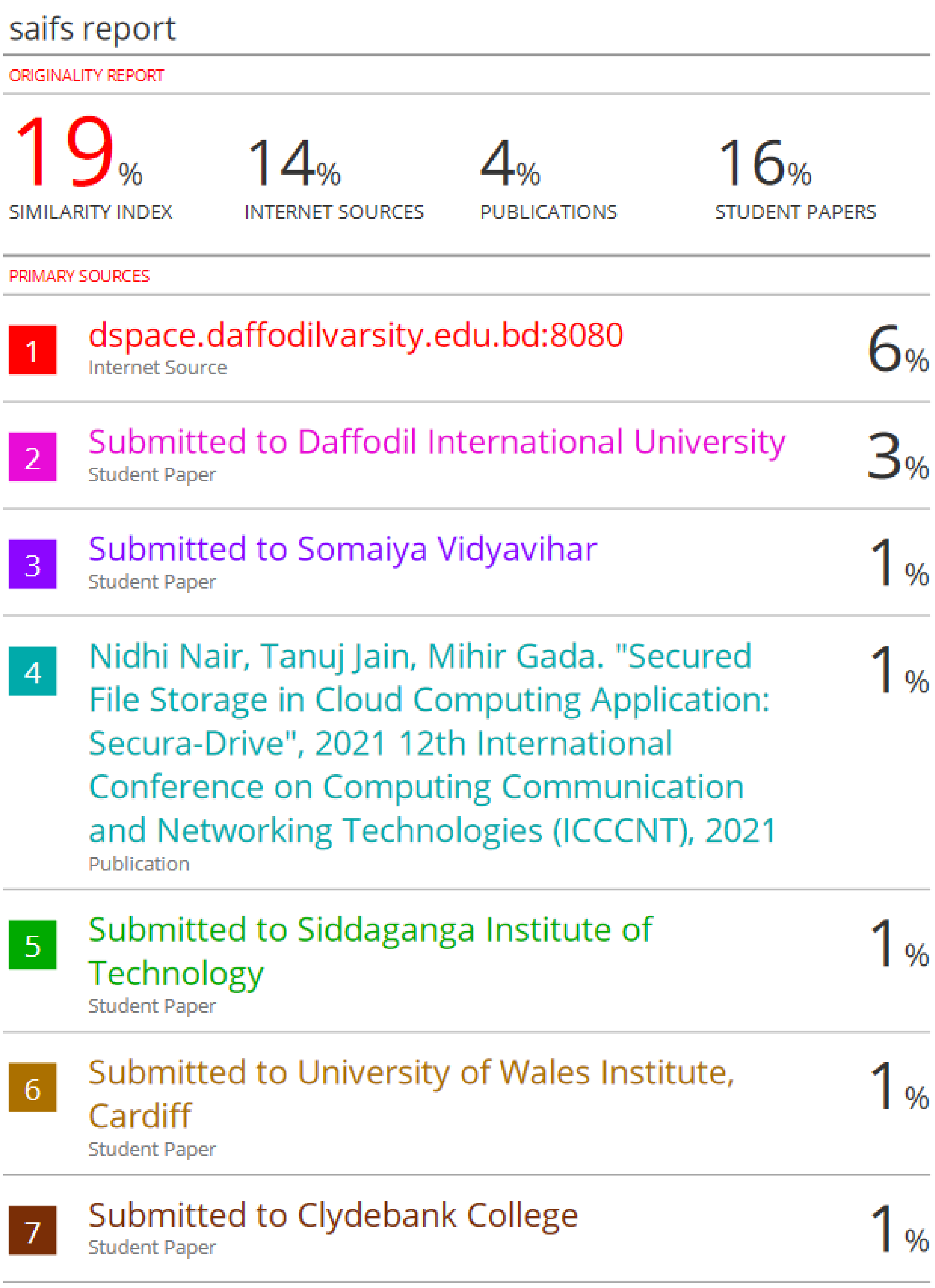
All the following libraries were used while implementing the Secure Drive Application :

- asgiref==3.3.1
- certifi==2020.12.5
- cffi==1.14.5
- chardet==4.0.0
- click==7.1.2
- cryptography==3.4.4
- Django==3.0.5
- djangorestframework==3.12.2

- djongo==1.3.4
- dlib==19.22.0
- dnspython==2.1.0
- docopt==0.6.2
- face-recognition==1.3.0
- face-recognition-models==0.3.0
- idna==2.10
- numpy==1.20.2
- opencv-python==4.5.1.48
- Pillow==8.2.0
- pipreqs==0.4.10
- pycparser==2.20
- pymongo==3.11.3
- pytz==2021.1
- requests==2.25.1
- sqlparse==0.2.4
- urllib3==1.26.3
- yarg==0.1.9

References:

[1] National Bureau of Standards, “Data Encryption Standard," FIPS Publication #46, NTIS, Apr. 1977.
[2] Mohammad Reza, Abbasy Mahd ,i Sharifi Mohammadreza “Cryptographic File System: Easy and Reliable? " International Journal of Information and Electronics Engineering Vol 3. Nov 2013
[3] Joachim von zur Gathen ,CryptoSchool,2015
[4] Scott Vanstone, Handbook of Applied Cryptography, 1996.
[5] Al Sweigart ,Hacking Secret Ciphers with Python
A beginner's guide to cryptography and computer programming with Python,2013
[6] Techtarger , available at << https://www.techtarget.com/searchsecurity/definition/cipher-block-chaining>>, last accessed on 27-12-2022 22:48.
[7] Geeksforgeeks, available at << https://www.geeksforgeeks.org/cryptography-and-its-types/>>, last accessed on 26-12-2022 22:48.



saifs report

ORIGINALITY REPORT

19%	14%	4%	16%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	6%
2	Submitted to Daffodil International University Student Paper	3%
3	Submitted to Somaiya Vidyavihar Student Paper	1%
4	Nidhi Nair, Tanuj Jain, Mihir Gada. "Secured File Storage in Cloud Computing Application: Secura-Drive", 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021 Publication	1%
5	Submitted to Siddaganga Institute of Technology Student Paper	1%
6	Submitted to University of Wales Institute, Cardiff Student Paper	1%
7	Submitted to Clydebank College Student Paper	1%