# SECURE FILE SHARING SYSTEM PROJECT

Topic: Sharing passwords and sensitive information with the loved ones.

Team members: Saakshara Kanyadara (SXK210041), Anish Joshi (AXJ200101), Sandra Jayakumar (SXJ210016), Sherin Jayakumar (SXJ210018), Rishitha Chitteti (RXC210003)

## MAJOR STEPS OF THE IMPLEMENTATION

Python was our go-to programming language because of its flexibility and availability of a vast number of libraries. The following were the major steps involved in the implementation:

1. We started the project by creating an account on Google API console and got the credentials for running the Google Drive API. We followed a very interesting blog by Matt Button (source provided below in the Reference section) to initialize the flask API. We also went through Google's QuickStart guide to learn about authentication, downloading, uploading, etc.
2. We used the "cryptography" library of Python to implement AES-GCM and the "scrypt" from the PyCryptodome library of Python to implement salting.
3. The Shamir Secret Sharing scheme was taken care by the "Shamir.js" (Source provided in Reference section)

First, the user navigates to http://localhost:8040/google/login and signs in with the account using OAuth. After logging in, the home page appears where the user can view the files already uploaded. Additionally, the user can upload more files as well. The user must enter a password before uploading the file. After uploading the file, the user has options to view/edit file, view the uploaded file or to delete the file. The Shamir's secret sharing scheme comes into the picture. The secret keys are generated and shared with the file URL. As per the project description, for this project, we kept the threshold as 2 and the no of secret keys as 3. The secret shares are printed in hexadecimal format so that it's easy for the user to copy it.

## DIFFICULTIES FACED DURING THE IMPLEMENTATION

1. Understanding AES-GCM and searching the most useful library for implementing AES-GCM and Shamir.js for our project description. We tried to use "PyCryptodome" as well but due to several issues, we shifted to "cryptography".
2. Understanding the implementation of Shamir.js and how to integrate it with our main.js file.
3. To check for unauthorized modification to the file contents. We had to use different functions to test the tags such as verify(), finalize() and encrypt_and_digest(). Finally, we were able to use the "aad" argument in the encrypt() and decrypt() function of the AES-GCM module of the "cryptography" library.
4. Due to unknown reasons, the CSS wasn't getting integrated with the HTML in the front-end.

## DIVISION OF LABOR

Saakshara Kanyadara: Worked on integrating all the components together, debugging the code logs and setting permissions in the Google API console.

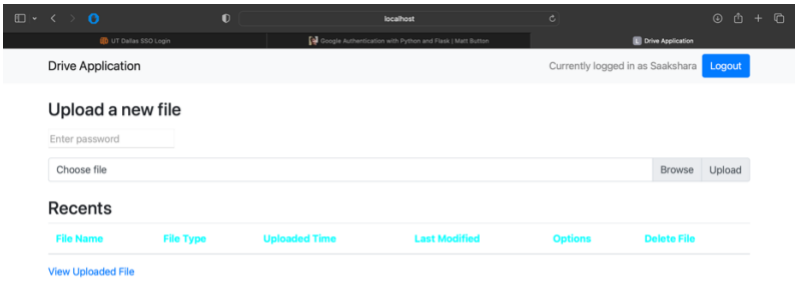
Anish Joshi: Worked on implementing AES-GCM encryption and decryption functionalities in Python, writing the project report and debugging.

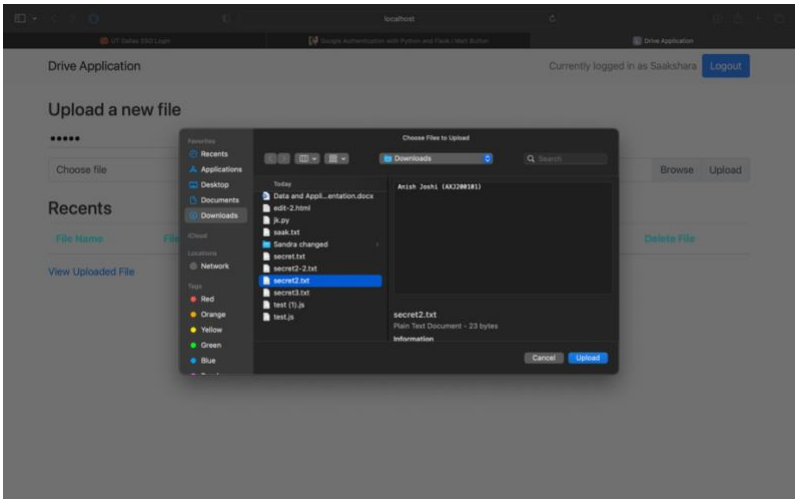
Sandra Jayakumar: Worked on the frond-end development of the application using HTML, CSS and JavaScript, and Debugging the code.

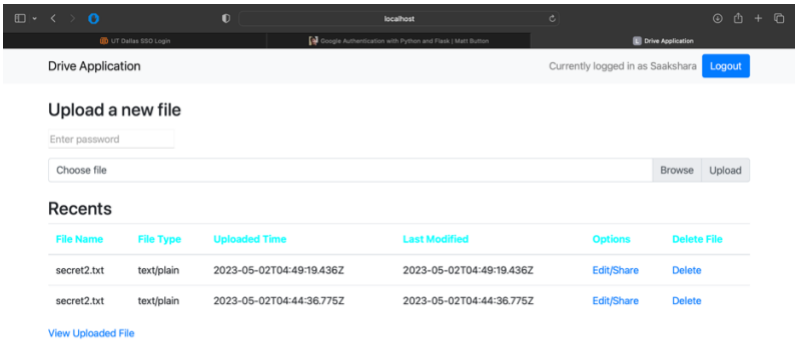
Sherin Jayakumar: Worked on the frond-end development of the application using HTML/CSS and JavaScript and the python implementation of Flask.

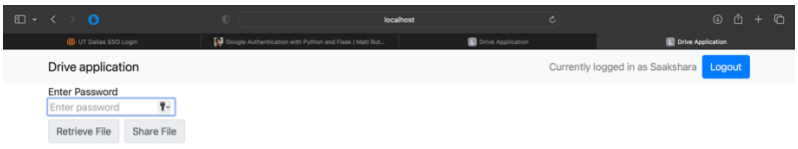
Rishitha Chitteti: Worked on integrating all the components together and creating the project report.

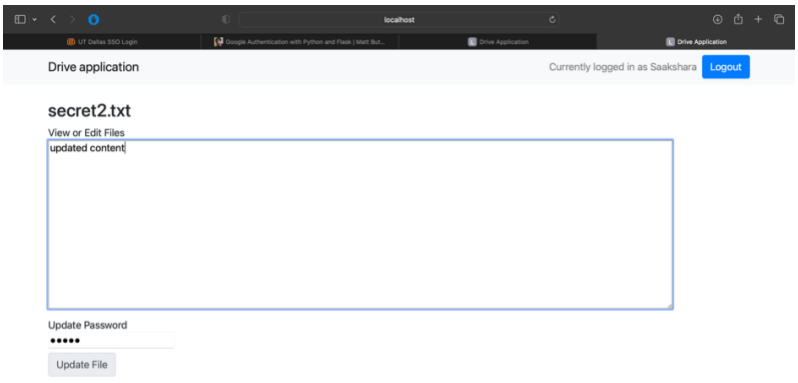
## TECHNOLOGIES USED AND SCREENSHOTS

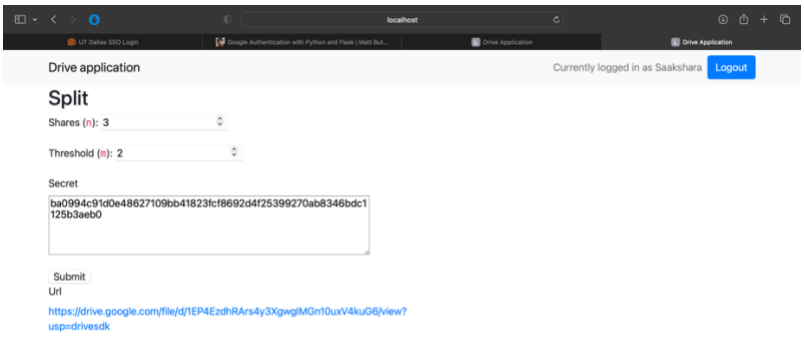


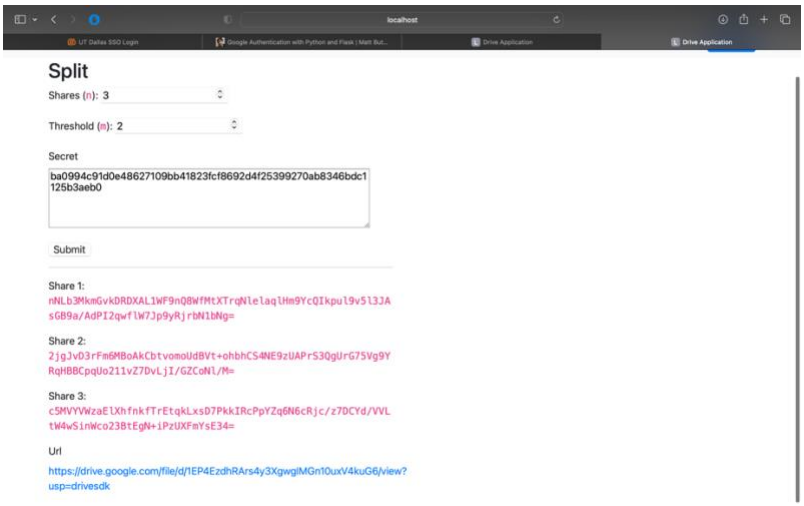




localhost
UT Dallas SSO Login
Google Authentication with Python and Flask | Matt Button
Drive Application
Drive Application
Currently logged in as Saakshara
Logout
Upload a new file
Enter password
Choose file
Browse
Upload
Recents
File Name
File Type
Uploaded Time
Last Modified
Options
Delete File
secret2.txt
text/plain
2023-05-02T04:49:19.436Z
2023-05-02T04:49:19.436Z
Edit/Share
Delete
secret2.txt
text/plain
2023-05-02T04:44:36.775Z
2023-05-02T04:44:36.775Z
Edit/Share
Delete
View Uploaded File


localhost
UT Dallas SSO Login
Google Authentication with Python and Flask | Matt But...
Drive Application
Drive Application
Drive application
Currently logged in as Saakshara
Logout
Enter Password
Enter password
Retrieve File
Share File


localhost
UT Dallas SSO Login
Google Authentication with Python and Flask | Matt But...
Drive Application
Drive Application
Drive application
Currently logged in as Saakshara
Logout
secret2.txt
View or Edit Files
updated content
Update Password
Update File




External and built-in libraries: AES-GCM from cryptography, scrypt from PyCryptodome and Shamir.js
Front-end: HTML, CSS and JavaScript
Back-end: Python with Flask
File sharing system: Google Drive

**REFERENCE**

1. https://www.mattbutton.com/2019/01/05/google-authentication-with-python-and-flask/
2. https://github.com/unusualbob/shamirJS
3. https://cryptography.io/en/latest/hazmat/primitives/aead/
4. https://pycryptodome.readthedocs.io/en/latest/src/api.html