SECURE FILE SHARING SYSTEM

NAME: SMRIDHI GERA

TASK 3: SECURE FILE SHARING SYSTEM

PROGRAM: FUTURE INTERNS- CYBERSECURITY INTERNSHIP

DATE: OCTOBER 2025

TOOLS USED : PYTHON 3.10+,FLASK FRAMEWORK,PYCRYPTODOME,BROWSER

INTRODUCTION

The Secure File Sharing System is a web-based application that allows users to upload, download, and delete files securely. All uploaded files are encrypted using AES-256 encryption before being stored, ensuring data confidentiality and security. This project demonstrates the use of Flask for backend development, cryptography for encryption, and modern frontend technologies (HTML, CSS, Bootstrap) for a responsive user interface.

OBJECTIVES

- To build a secure platform for file sharing.
- To encrypt files using AES-256 (GCM mode).
- To enable upload, download, and delete functionality.
- To design a responsive and user-friendly interface.

SYSTEM REQUIREMENTS

- Python 3.10+
- Flask
- PyCryptodome □ Browser (Firefox/Chrome)
- OS: Windows/Linux/Kali Linux

IMPLEMENTATION STEPS

1. Environment Setup

- Created project folder: secure-file-sharing
- Initialized virtual environment: python -m venv venv

source venv/bin/activate # Windows: venv\Scripts\activate

▯ Installed dependencies: pip install Flask pycryptodome

2. AES Key Generation

- Generated a 32-byte AES key and saved as key.key.

3. Backend (Flask App)

▯ Implemented routes:

- / → Homepage (list files)
- /upload → Upload & encrypt file
- /download/<filename> → Decrypt & download file
- /delete/<filename> → Delete file

4. Frontend (HTML + Bootstrap)

▯ Responsive UI with:

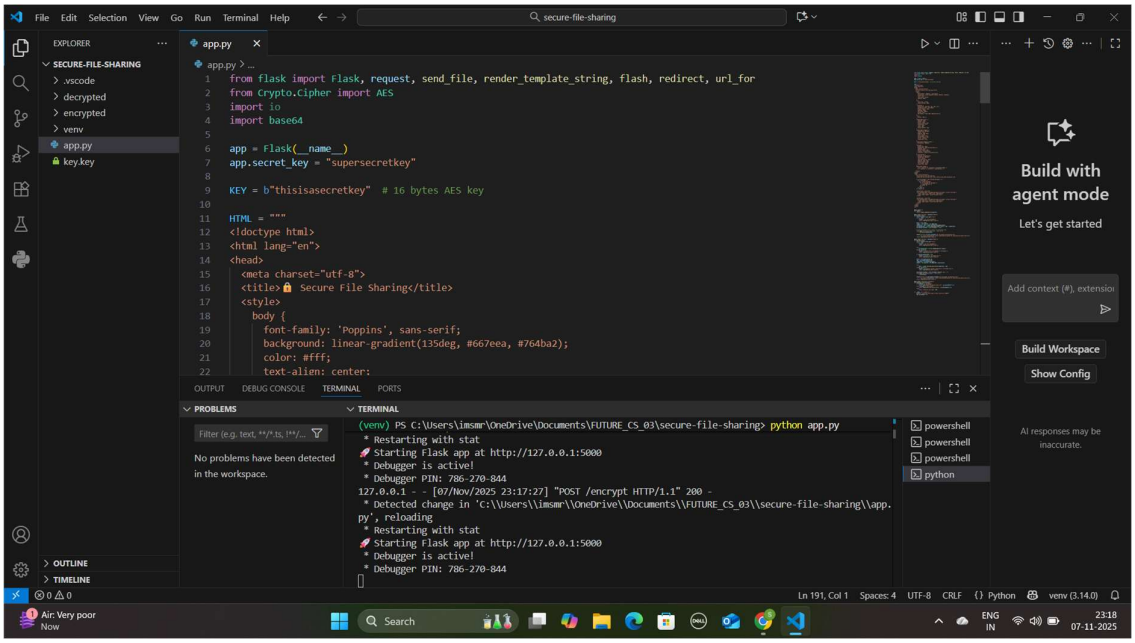
- Upload form
- File listing section
- Download/Delete buttons
- Notification messages

5. Encryption & Decryption

- Used AES-GCM for confidentiality and integrity.
- Encrypted files stored with .enc extension.
- Decrypted files downloaded with original name.

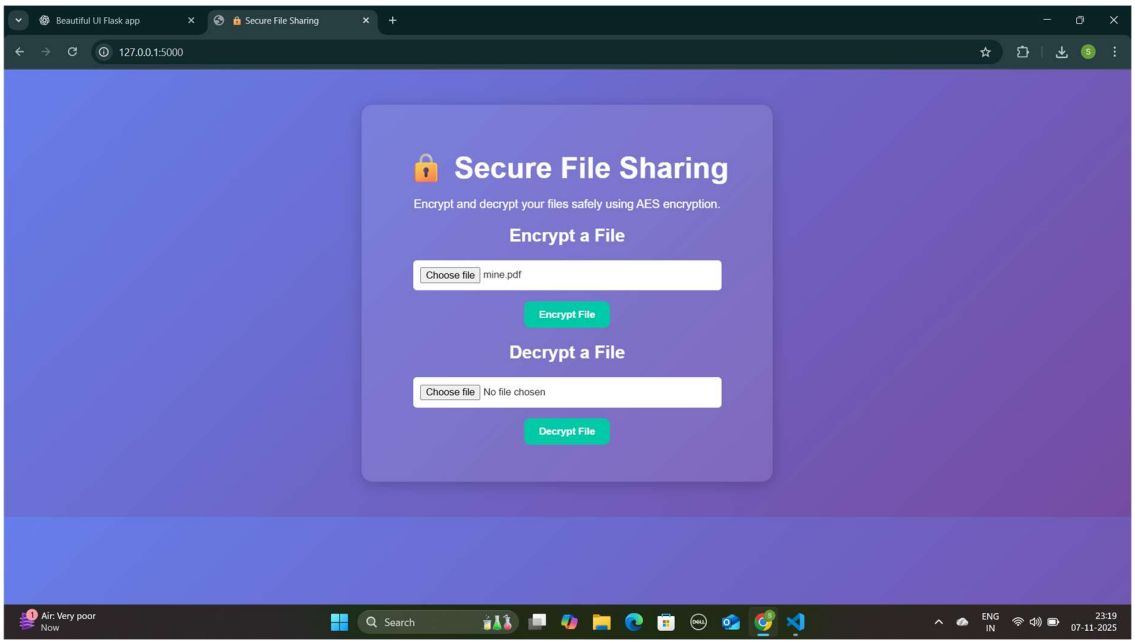
OUTPUT SCREENSHOTS

1. Flask Server Running:

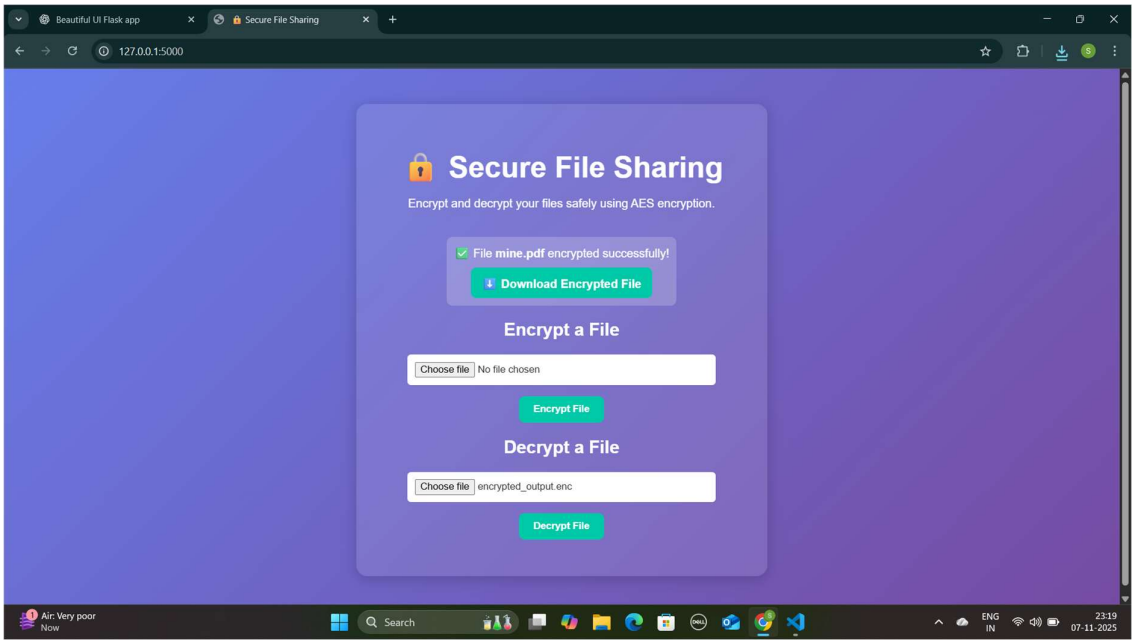


2. File Upload & Listing:

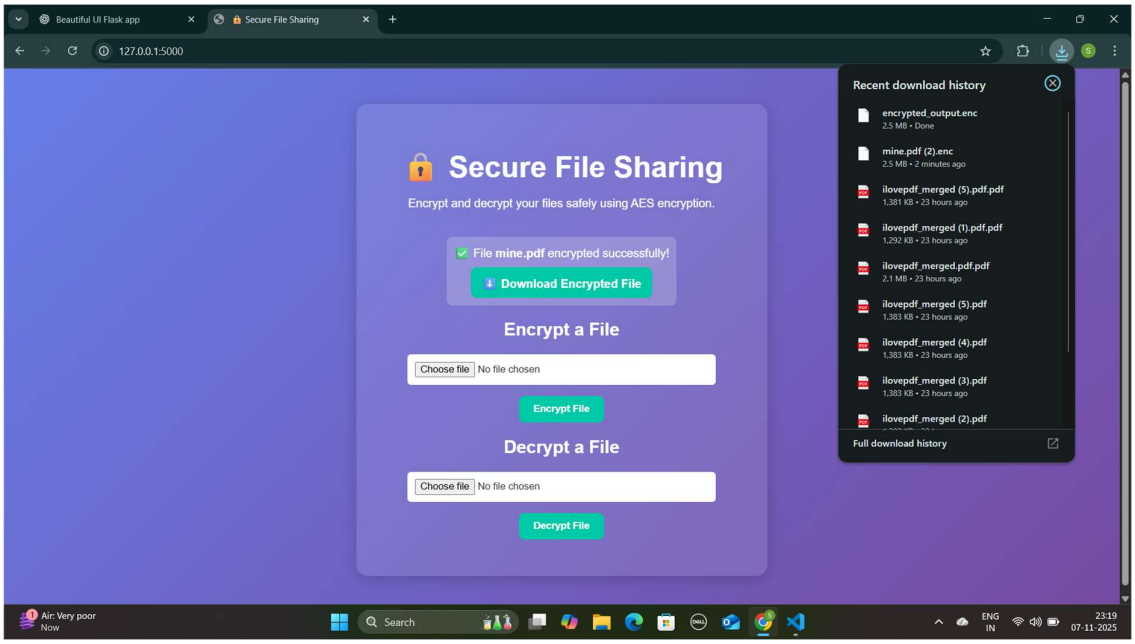
   ENCRYPT FILE

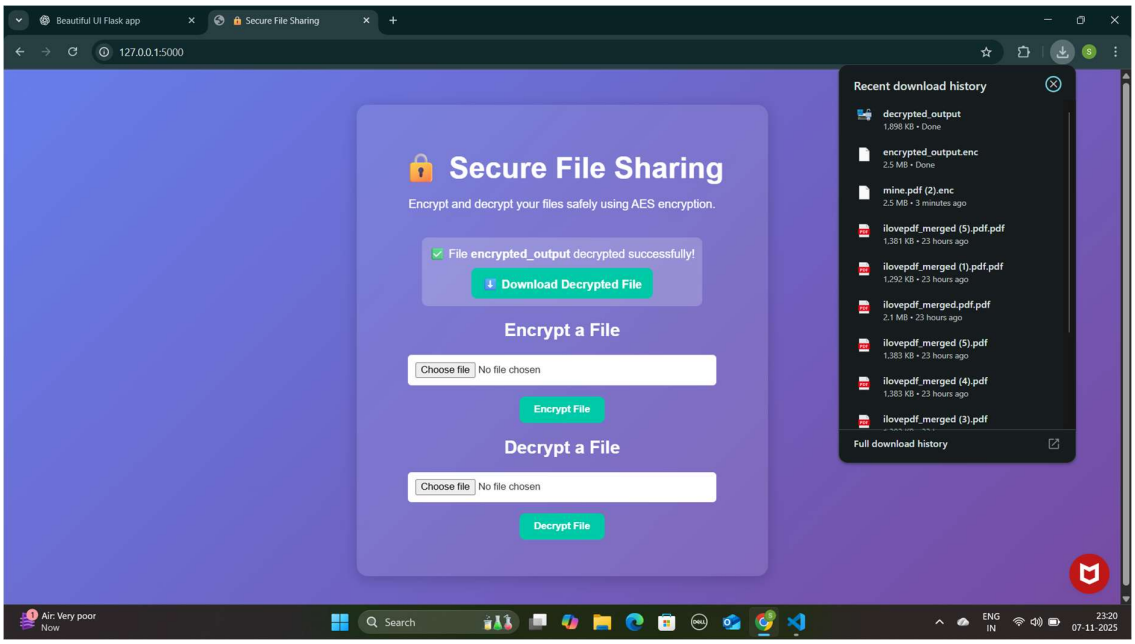

Beautiful UI Flask app
Secure File Sharing
127.0.0.1:5000
Secure File Sharing
Encrypt and decrypt your files safely using AES encryption.
Encrypt a File
Choose file
mine.pdf
Encrypt File
Decrypt a File
Choose file
No file chosen
Decrypt File
Air: Very poor
Now
Search
ENG
IN
23:19
07-11-2025

DECRYPT


Beautiful UI Flask app
Secure File Sharing
127.0.0.1:5000
Secure File Sharing
Encrypt and decrypt your files safely using AES encryption.
File mine.pdf encrypted successfully!
Download Encrypted File
Encrypt a File
Choose file
No file chosen
Encrypt File
Decrypt a File
Choose file
encrypted_output.enc
Decrypt File
Air: Very poor
Now
Search
ENG
IN
23:19
07-11-2025

3. File Download encryption/Decryption success:





RESULTS

The system successfully allows secure uploading, downloading, and deleting of files while ensuring confidentiality through AES-256 encryption.

CONCLUSION

This project demonstrates how to build a secure file sharing system using Flask and AES encryption. It ensures data confidentiality during upload, storage, and download. The system is simple, effective, and ready for future enhancements.

FUTURE ENHANCEMENTS

- Add user authentication (Login/Signup)
- Implement file size limits

- Support cloud storage (AWS/GCP/Azure)
- Add audit logs for file activity tracking

REFERENCES

1. https://flask.palletsprojects.com/en/stable/
2. https://pycryptodome.readthedocs.io/en/latest/
3. https://docs.python.org/3/