



SECURE FILE SHARING SYSTEM REPORT

NAME: IBE KINGSLEY

TASK 3: SECURE FILE SHARING SYSTEM

PROGRAM: FUTURE INTRENS CYBERSECURITY INTERNSHIP

DATE: SEPTEMBER 2025

TOOLS USED: PYTHON 3.10+, Flask Framework, HTML5, CSS3 (Bootstrap)
ASE Encryption (via PyCryptodome), LINUX (Kali).

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) for a responsive UI

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: Kali Linux

IMPLEMENTATION STEPS

1. Environment Setup

- Created project folder: `secure-file-sharing`
- Initialized virtual environment:
python3 -m venv venv
source venv/bin/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)

- Routes implemented:

/ → 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

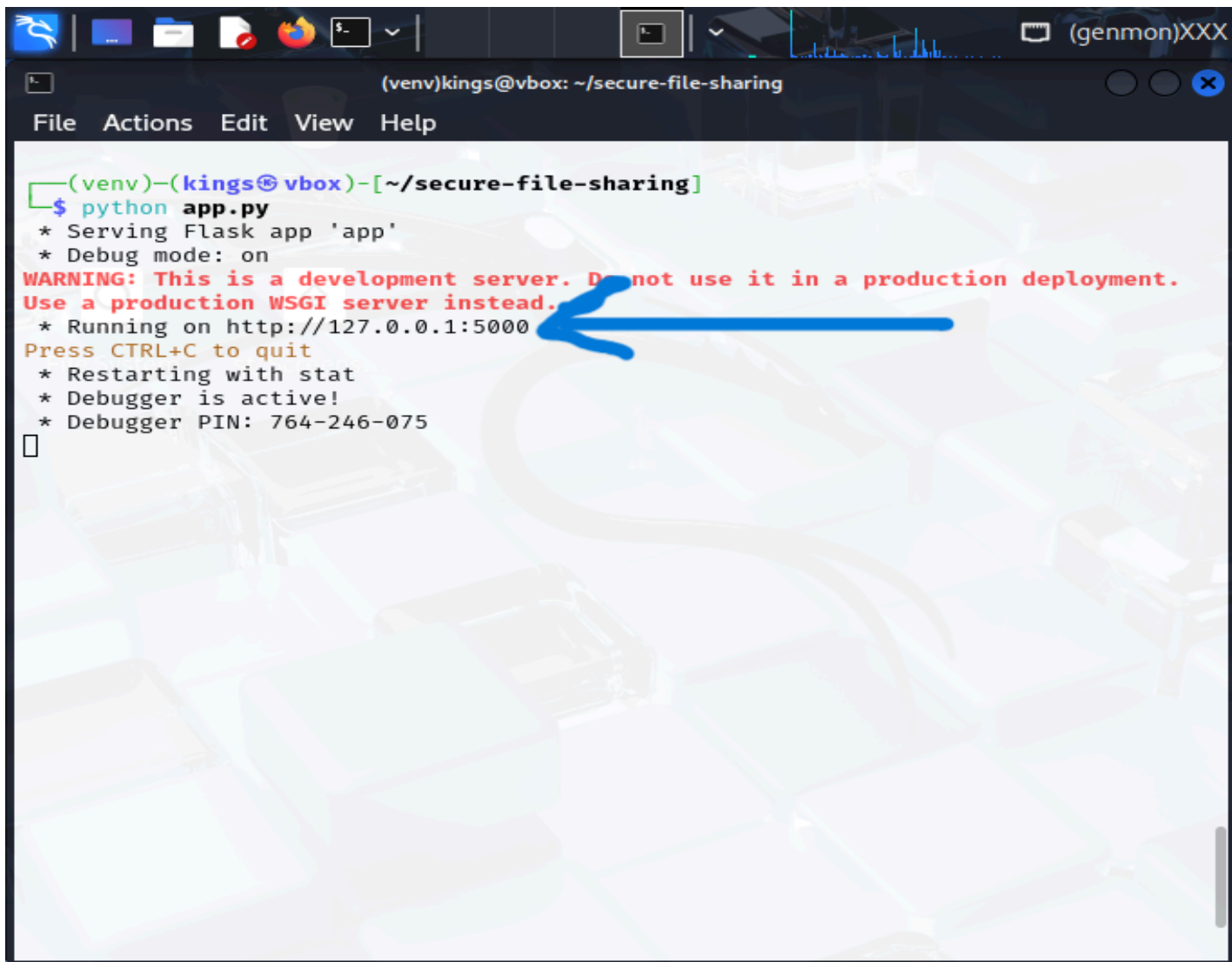
Notification messages

5. Encryption & Decryption

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

OUTPUT SCREENSHOTS

1. Flask server running

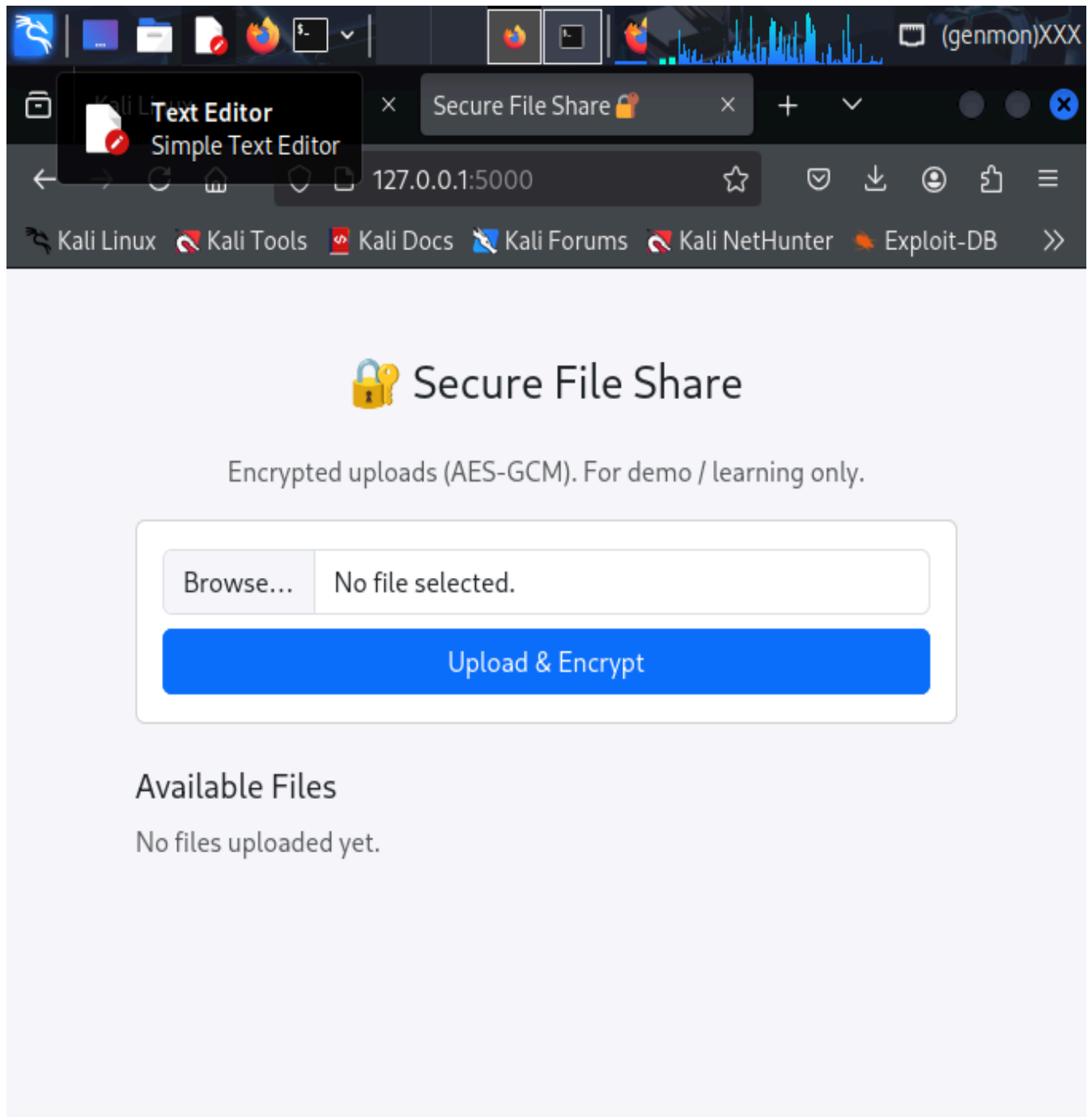


```
(venv)kings@vbox: ~/secure-file-sharing
File Actions Edit View Help

(venv)-(kings@vbox)-[~/secure-file-sharing]
$ python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 764-246-075
```

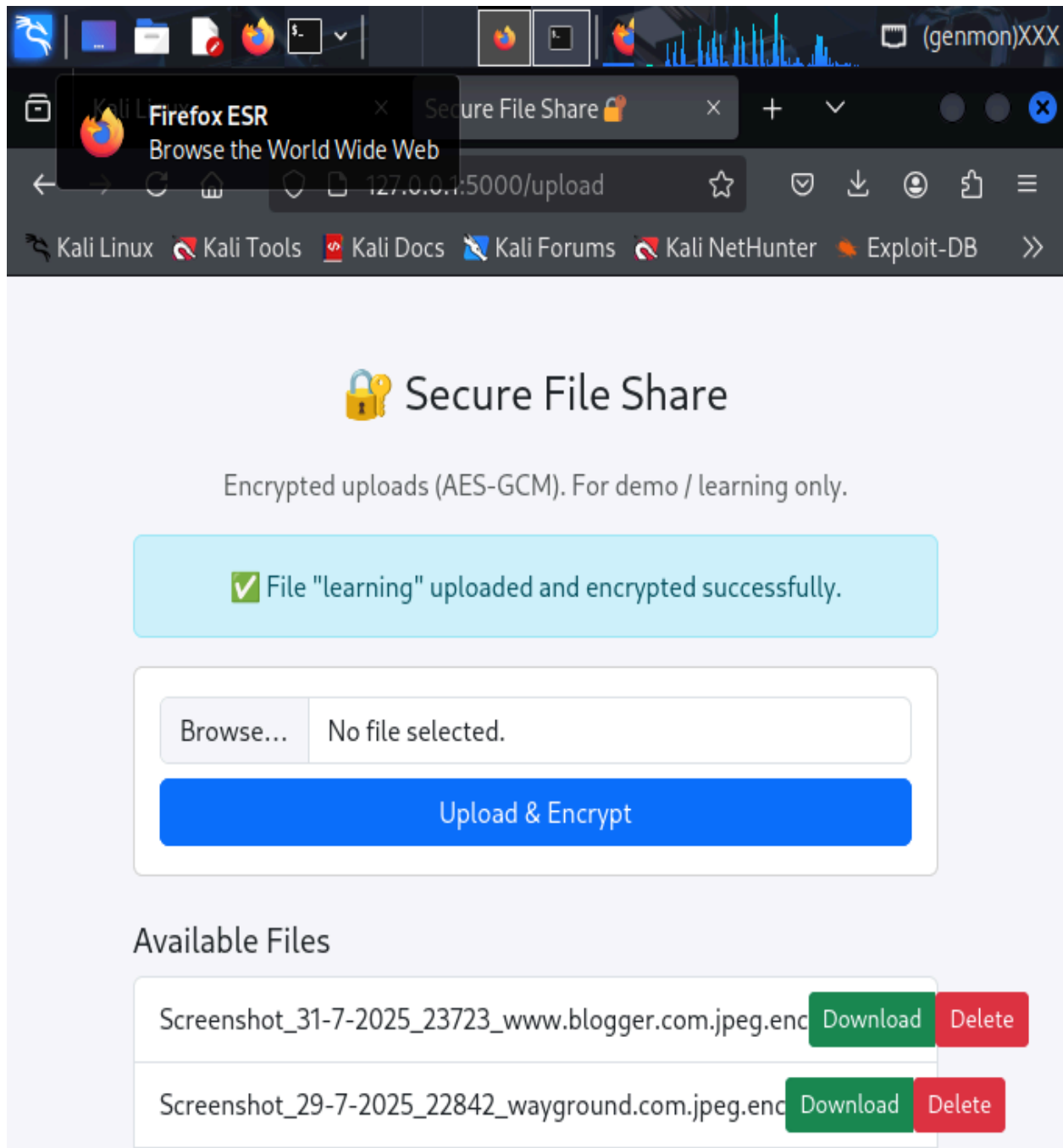
Screenshot1_ServerStart.png

2. Homepage with Upload and encrypt form



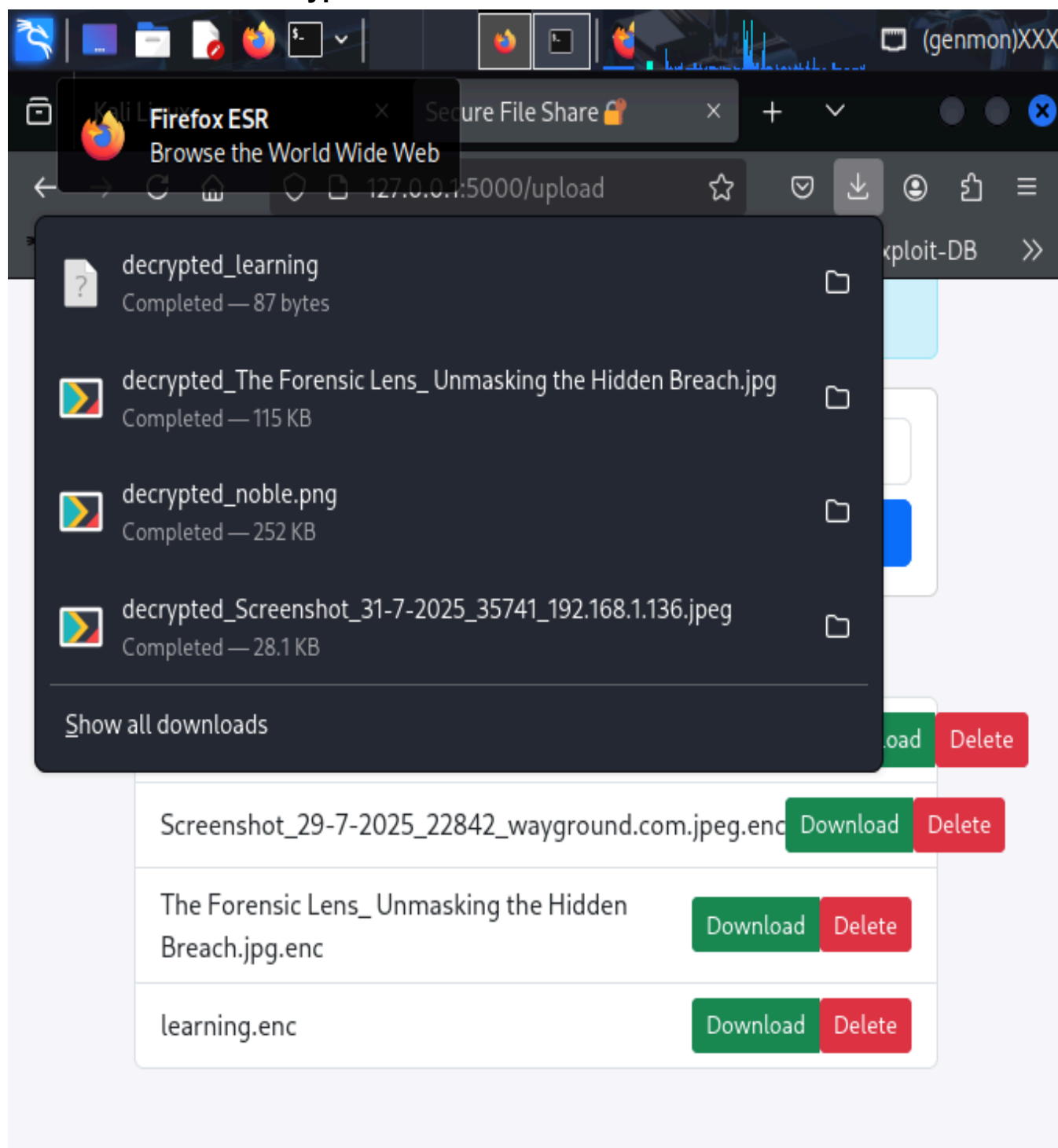
Screenshot2_HomepageStyled.png

3. Upload success message



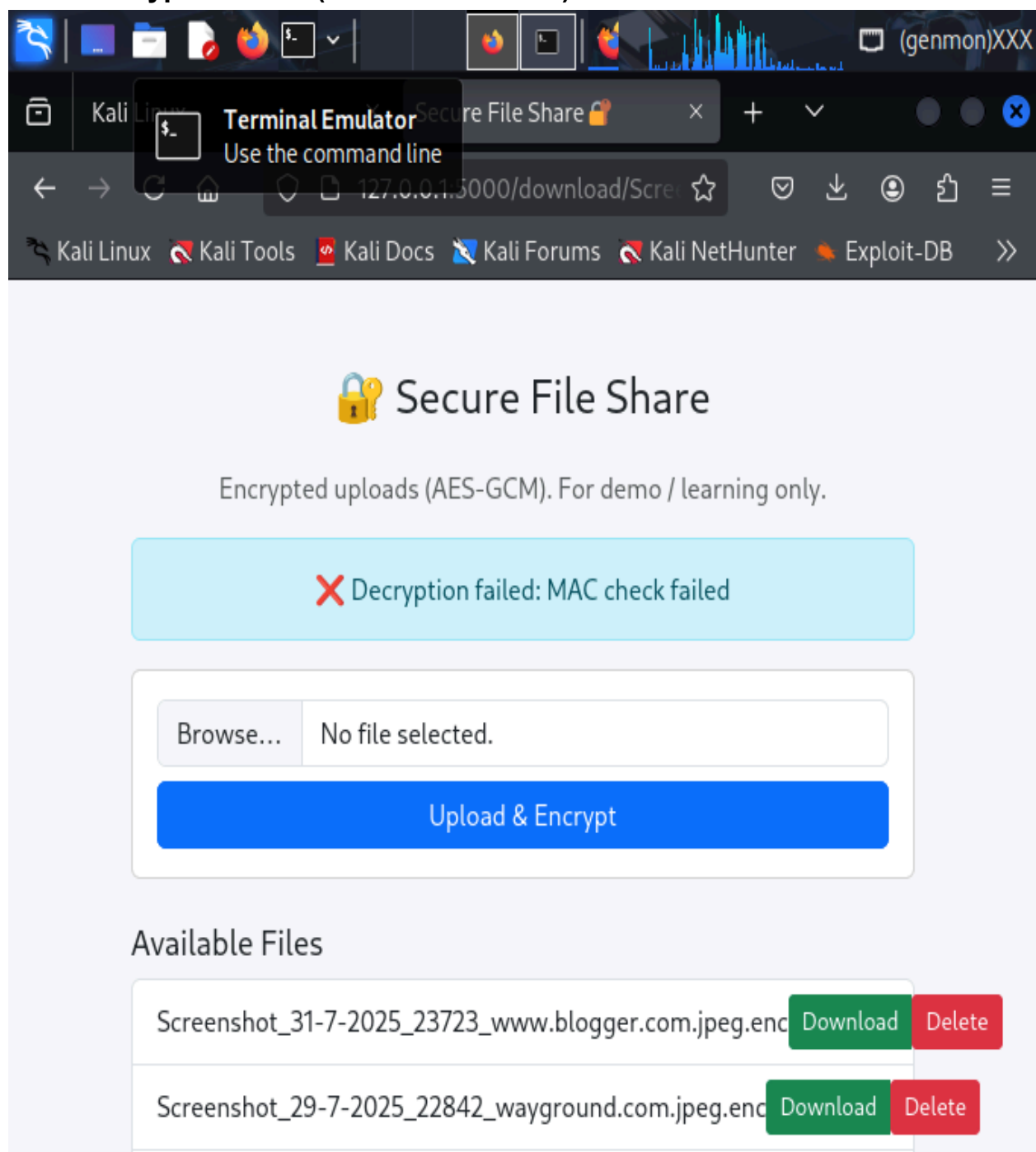
Screenshot3_UploadSuccess.png

4. Downloaded decrypted file



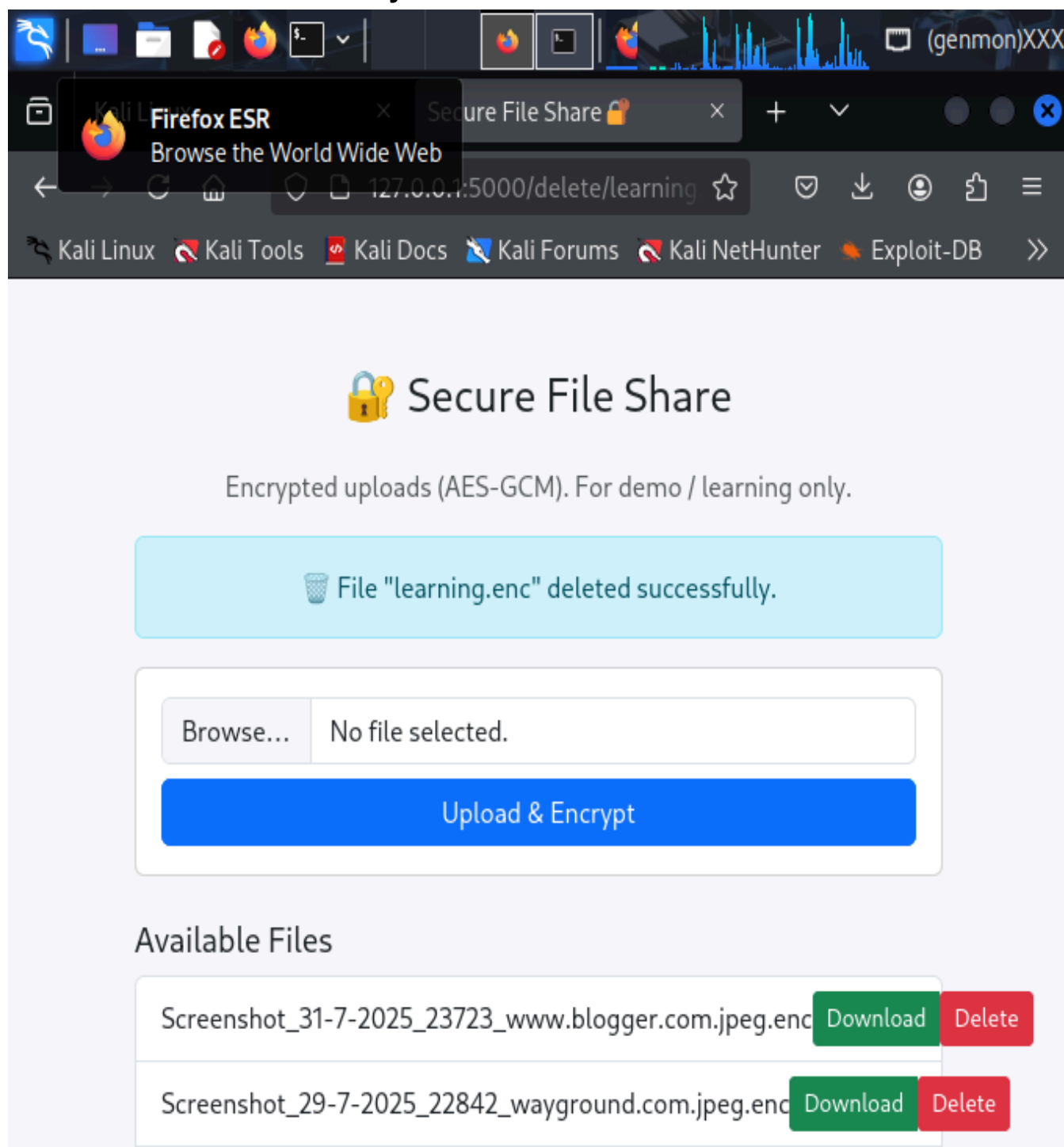
Screenshot4_DownloadFile.png

5. Decryption error (MAC check failed)



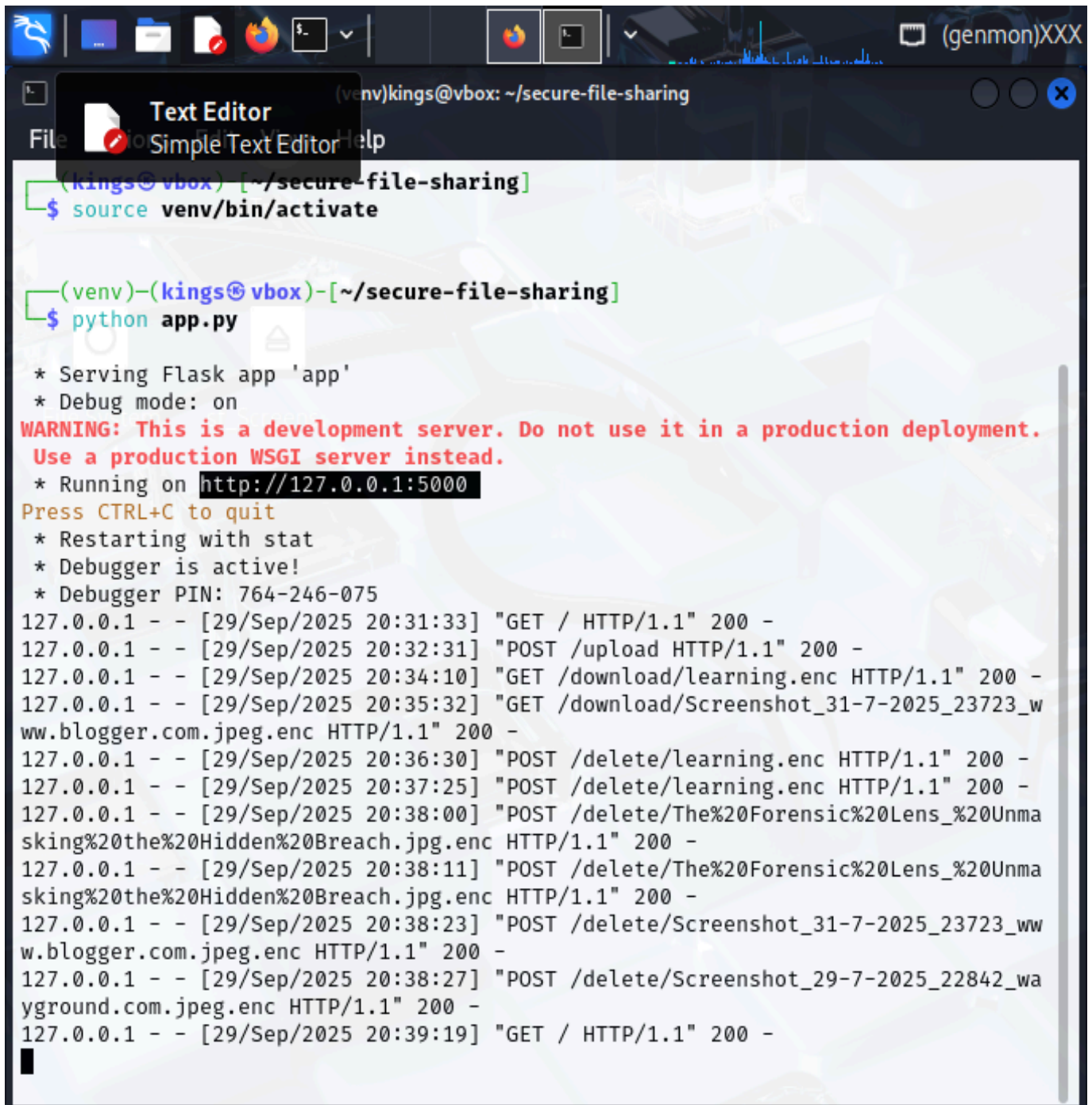
Screenshot5_DecryptionError.png

6. File deleted successfully



Screenshot6_DeleteSuccess.png

7. Flask server Running



```
(venv)kings@vbox: ~/secure-file-sharing
$ source venv/bin/activate

(venv)-(kings@vbox)-[~/secure-file-sharing]
$ python app.py

* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 764-246-075
127.0.0.1 - - [29/Sep/2025 20:31:33] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [29/Sep/2025 20:32:31] "POST /upload HTTP/1.1" 200 -
127.0.0.1 - - [29/Sep/2025 20:34:10] "GET /download/learning.enc HTTP/1.1" 200 -
127.0.0.1 - - [29/Sep/2025 20:35:32] "GET /download/Screenshot_31-7-2025_23723_ww
w.blogger.com.jpeg.enc HTTP/1.1" 200 -
127.0.0.1 - - [29/Sep/2025 20:36:30] "POST /delete/learning.enc HTTP/1.1" 200 -
127.0.0.1 - - [29/Sep/2025 20:37:25] "POST /delete/learning.enc HTTP/1.1" 200 -
127.0.0.1 - - [29/Sep/2025 20:38:00] "POST /delete/The%20Forensic%20Lens_%20Unma
sking%20the%20Hidden%20Breach.jpg.enc HTTP/1.1" 200 -
127.0.0.1 - - [29/Sep/2025 20:38:11] "POST /delete/The%20Forensic%20Lens_%20Unma
sking%20the%20Hidden%20Breach.jpg.enc HTTP/1.1" 200 -
127.0.0.1 - - [29/Sep/2025 20:38:23] "POST /delete/Screenshot_31-7-2025_23723_ww
w.blogger.com.jpeg.enc HTTP/1.1" 200 -
127.0.0.1 - - [29/Sep/2025 20:38:27] "POST /delete/Screenshot_29-7-2025_22842_wa
yground.com.jpeg.enc HTTP/1.1" 200 -
127.0.0.1 - - [29/Sep/2025 20:39:19] "GET / HTTP/1.1" 200 -
```

Screenshot7_serverrunning.png

RESULTS

- Files uploaded and encrypted successfully
- Files downloaded in decrypted form
- Files deleted securely
- UI is clean, responsive, and user-friendly

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

- Flask Documentation [\[1\]](#) [\[2\]](#)
- PyCryptodome Docs [\[1\]](#)
- Python Official Docs [\[1\]](#) [\[2\]](#)