Secure File Sharing System – Internship Project Report

# 1. Introduction

This project implements a secure file sharing web application using Flask (Python) and AES encryption to ensure confidentiality of files during upload and download. It simulates real-world scenarios where secure data sharing is critical.

# 2. Features

- Secure file upload & download  
- AES encryption for files at rest  
- Basic key management using .env file  
- User-friendly dashboard UI (Bootstrap 5)

# 3. Tools Used

- Python Flask  
- PyCryptodome (AES)  
- Bootstrap 5  
- GitHub for version control

# 4. Architecture

The architecture follows a simple client-server model:  
- User uploads → Flask server encrypts file → stores as .enc in uploads/ folder  
- User downloads → Flask decrypts → serves back original file

# 5. Security Measures

- AES-256 encryption ensures confidentiality of data  
- Environment variable-based key management using .env  
- No plaintext files are stored on the server  
- Encrypted files appear as unreadable gibberish if opened directly

# 6. How to Run

Step 1: Clone repo  
 git clone <your-repo-url>  
 cd secure-file-share  
  
Step 2: Create virtual environment  
 python -m venv venv  
 .\venv\Scripts\activate  
  
Step 3: Install dependencies  
 pip install -r requirements.txt  
  
Step 4: Run app  
 python app.py

# 7. Screenshots

Insert screenshots of:  
- Upload Page  
- File List with Encrypted Files  
- Decrypted Download Example

# 8. Conclusion

This project demonstrates how cryptography (AES) can be integrated into a real-world secure file-sharing system. It highlights secure coding practices and showcases a practical understanding of SOC-related secure development.