

SECURE FILE SHARING SYSTEM

Internship Project Report

Submitted by:

Adebowale Emmanuel Okikiola

Internship Program: Cybersecurity internship (Futureinterns)

Task: Task 3- secure file sharing system

Date of submission August 2025

CIN: FIT/JUL25/CS2572

Table of contents

1. Introduction

- 1.1 Background
- 1.2 Objective of the project

2. System setup & configuration

- 2.1 Tools & Technologies used
- 2.2 Installation Process
- 2.3 Folder Structure

3. Implementing of secure file sharing

- 3.1 Encryption Setup
- 3.2 File upload process
- 3.3 Storage & Security

4. Testing & Results

- 4.1 Uploading and Encryption Files
- 4.2 Accessing Uploaded Files
- 4.3 Screenshots

5. Challenges & Solution

6. Conclusion & Lessons Learned

7. References

1. Introduction

1.1 Background

In today's digital world, file sharing is essential for collaboration across organizations. However, sensitive data is often exposed to risks such as unauthorized access, data leaks, and as well Cyberattacks. A secure file sharing system provides a safer way to upload, store, and share files by integrating encryption and authentication mechanisms.

1.2 Objectives of the Project

The objective of this internship task was to design and implement a secure file-sharing system using python (Flask Framework). The system allows users to:

- Upload files securely
- Automatically encrypt files before storage
- Ensures files can only be accessed by authorized users

2. System setup & Configuration

2.1 Tools & Technologies Used

- Programming Language: python 3.11
- Framework: Flask
- Encryption Library: Cryptography (Fernet symmetric encryption)
- Virtual Environment: venv
- Operating System: Window 10/ Kali linux
- Text Editor/IDE: VS Code / Notepad

2.2 Installation process

1. Created a project folder `secure_file_share`
2. Set up a python virtual enviroment(`venv`)
3. Installed dependencies listed in `requirements.txt`:
 - Flask
 - Flask-Login
 - Cryptography
 - Boto 3(optional for cloud integration)
4. Created subfolders:
 - Templates (HTML templates)
 - Encrypted (for encrypted files)
5. Developed the main application file `app.py`

2.3 Folder structure

```
secure_file_share/  
|-----app.py  
|-----requirements.txt  
|-----secret.key  
|-----encrypted/  
|-----templates/  
|   |-----index.html  
|   |-----upload.html  
|   |-----files.html
```

3. Implementation of secure file sharing

3.1 Encryption Setup

- Used Fernet symmetric encryption from cryptography library.
- Generated a secret key (secret.key) to encrypt and decrypt files.

3.2 File Upload Process

- **User** selects a file on the upload page.
- File is encrypted before being saved into the encrypted/ folder.
- Original file is not stored to ensure data security.

3.3 Storage & Security

- Files are stored only in the encrypted format.
- Even if an attacker gains access to the folder, files remain unreadable without the encryption key.

4. Testing & Results

4.1 Uploading and Encrypting files

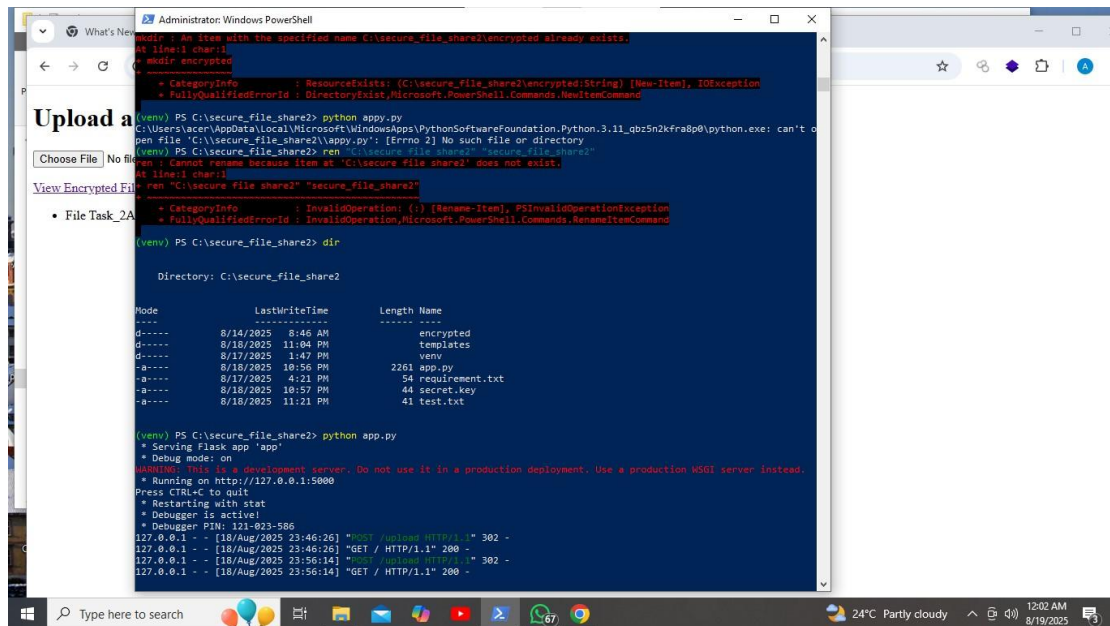
- Successfully uploaded text files, PDF files, and other documents.
- Verified decrypted content matched original uploaded files.

4.2 Accessing Uploading files

- Tested retrieval and decryption of files.
- Verified decrypted content matched original uploads files.

4.3 Screenshot

Runing Flask



```
Administrator: Windows PowerShell
mdir : An item with the specified name C:\secure_file_share2\encrypted already exists.
+ CategoryInfo          : ResourceExists (C:\secure_file_share2\encrypted:String) [New-Item], IOException
+ FullyQualifiedErrorId : DirectoryExist,Microsoft.PowerShell.Commands.NewItemCommand

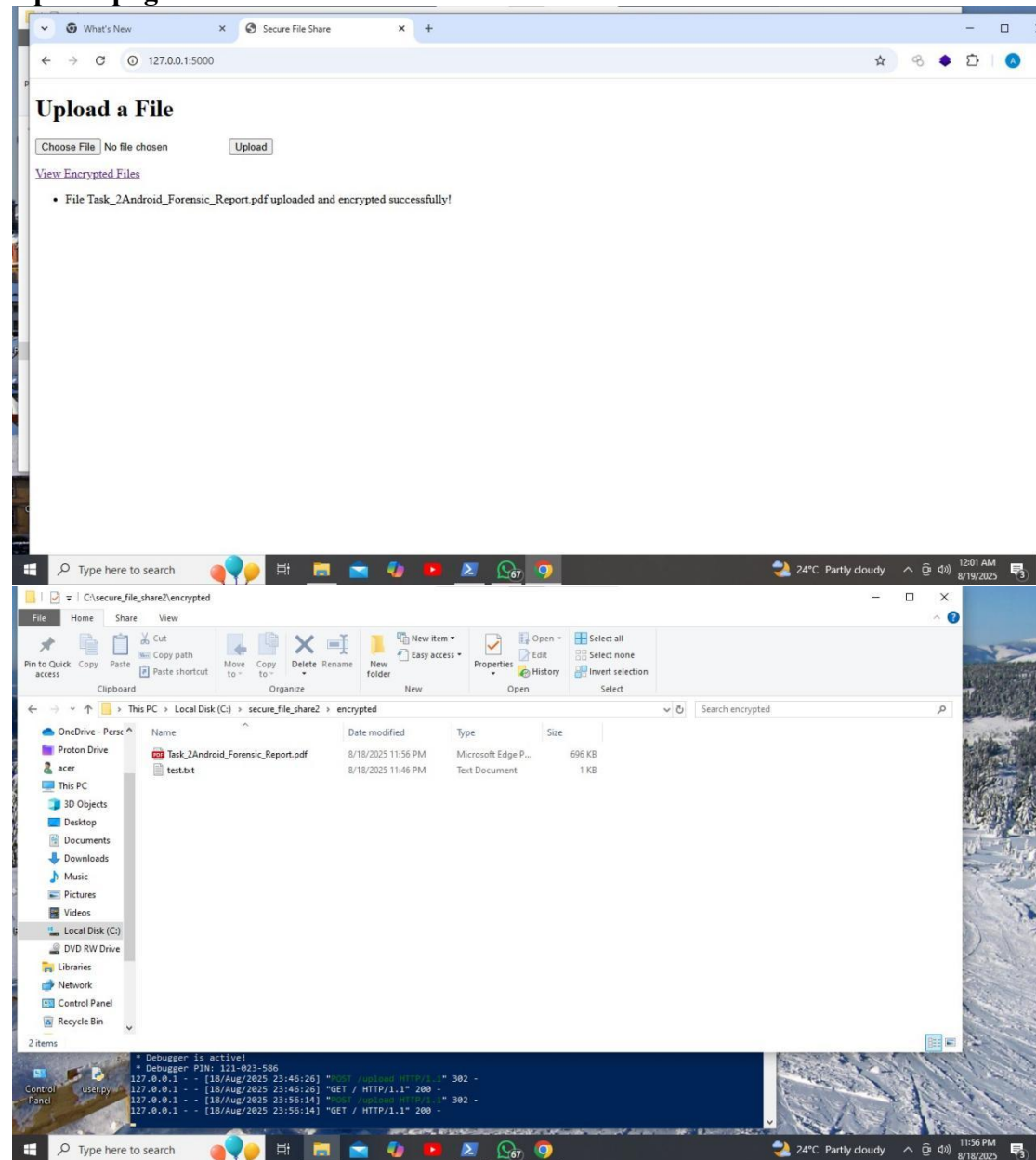
(venv) PS C:\secure_file_share2> python app.py
C:\Users\user\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\python.exe: can't o
pen file 'C:\secure_file_share2\app.py': [Errno 2] No such file or directory
(venv) PS C:\secure_file_share2> ren "C:\secure_file_share2" "secure_file_share2"
ren : Cannot rename because item at 'C:\secure_file_share2' does not exist.
(venv) PS C:\secure_file_share2> ren "C:\secure_file_share2" "secure_file_share2"
ren : Cannot rename because item at 'C:\secure_file_share2' does not exist.
(venv) PS C:\secure_file_share2> dir

Directory: C:\secure_file_share2

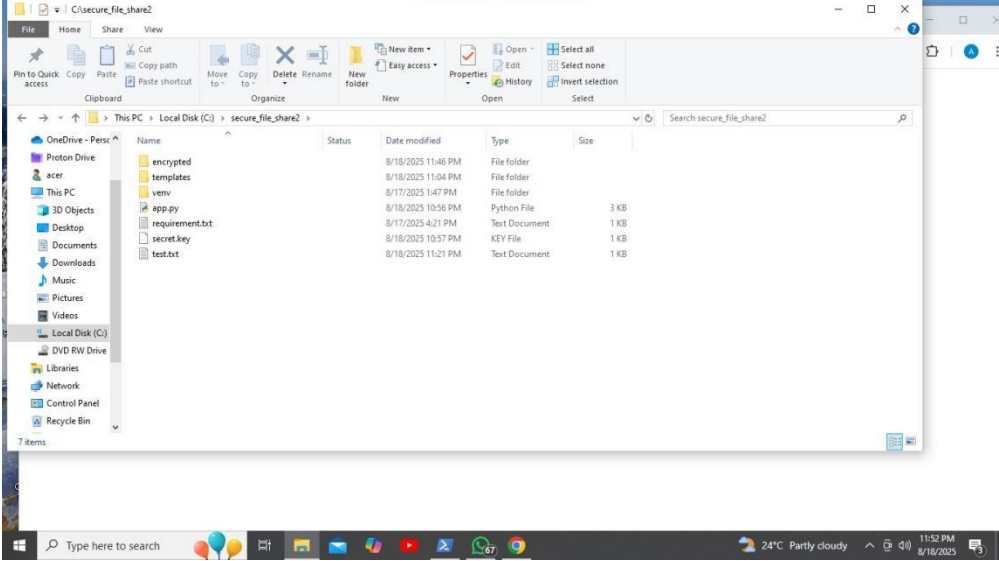
Mode                LastWriteTime         Length Name
-----
d-----      8/14/2025   8:46 AM             encrypted
d-----      8/18/2025  11:04 PM             templates
d-----      8/17/2025   1:47 PM             venv
-a-----      8/18/2025  10:56 PM           2261 app.py
-a-----      8/17/2025   4:21 PM            54 requirement.txt
-a-----      8/18/2025  10:57 PM            44 secret.key
-a-----      8/18/2025  11:21 PM            41 test.txt

(venv) PS C:\secure_file_share2> 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: 121-923-586
127.0.0.1 - - [18/Aug/2025 23:46:26] "POST /upload HTTP/1.1" 302 -
127.0.0.1 - - [18/Aug/2025 23:46:26] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [18/Aug/2025 23:56:14] "POST /upload HTTP/1.1" 302 -
127.0.0.1 - - [18/Aug/2025 23:56:14] "GET / HTTP/1.1" 200 -
```

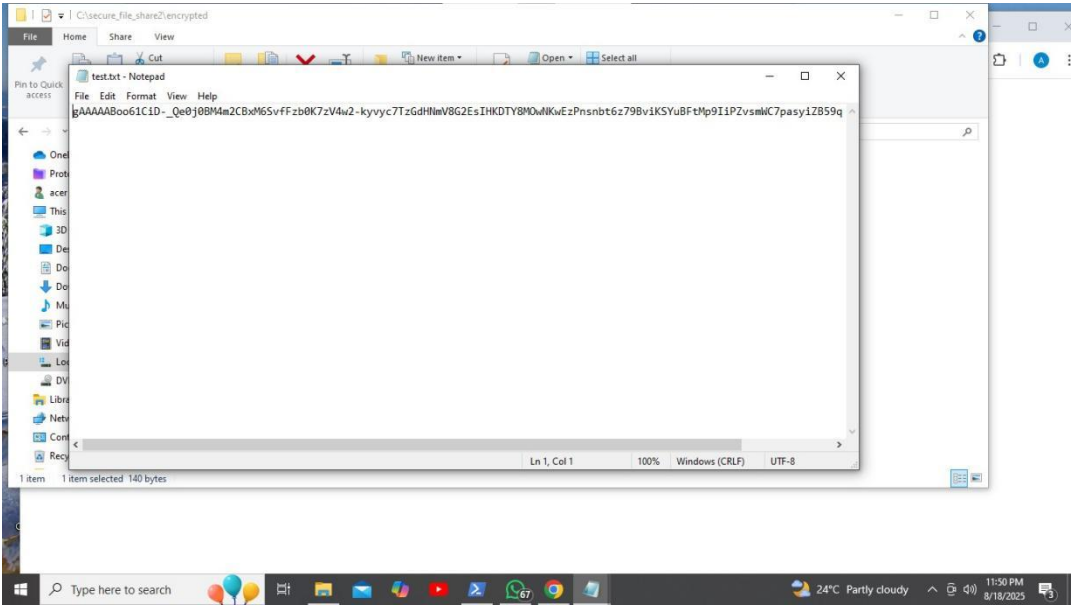
Upload page

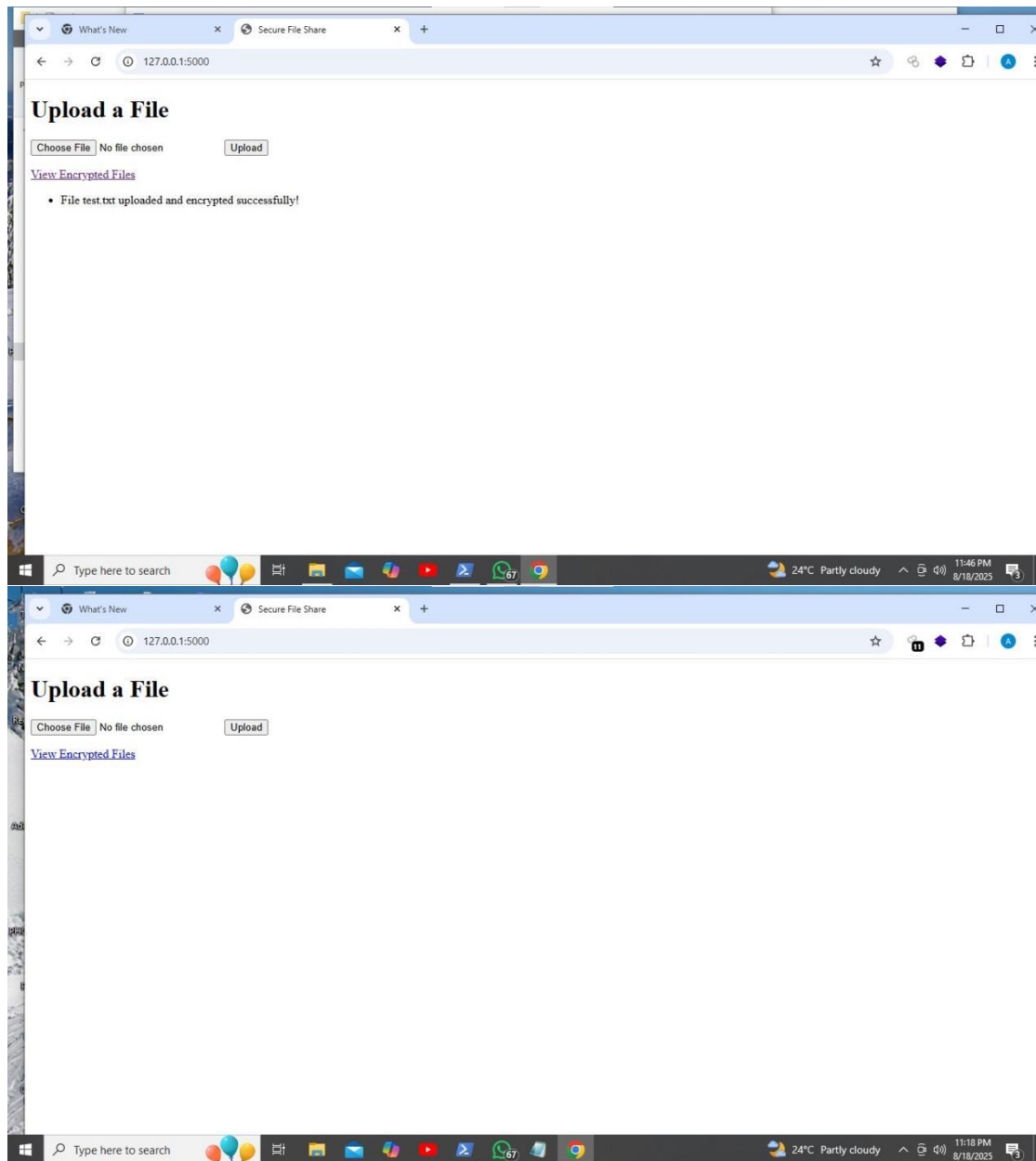


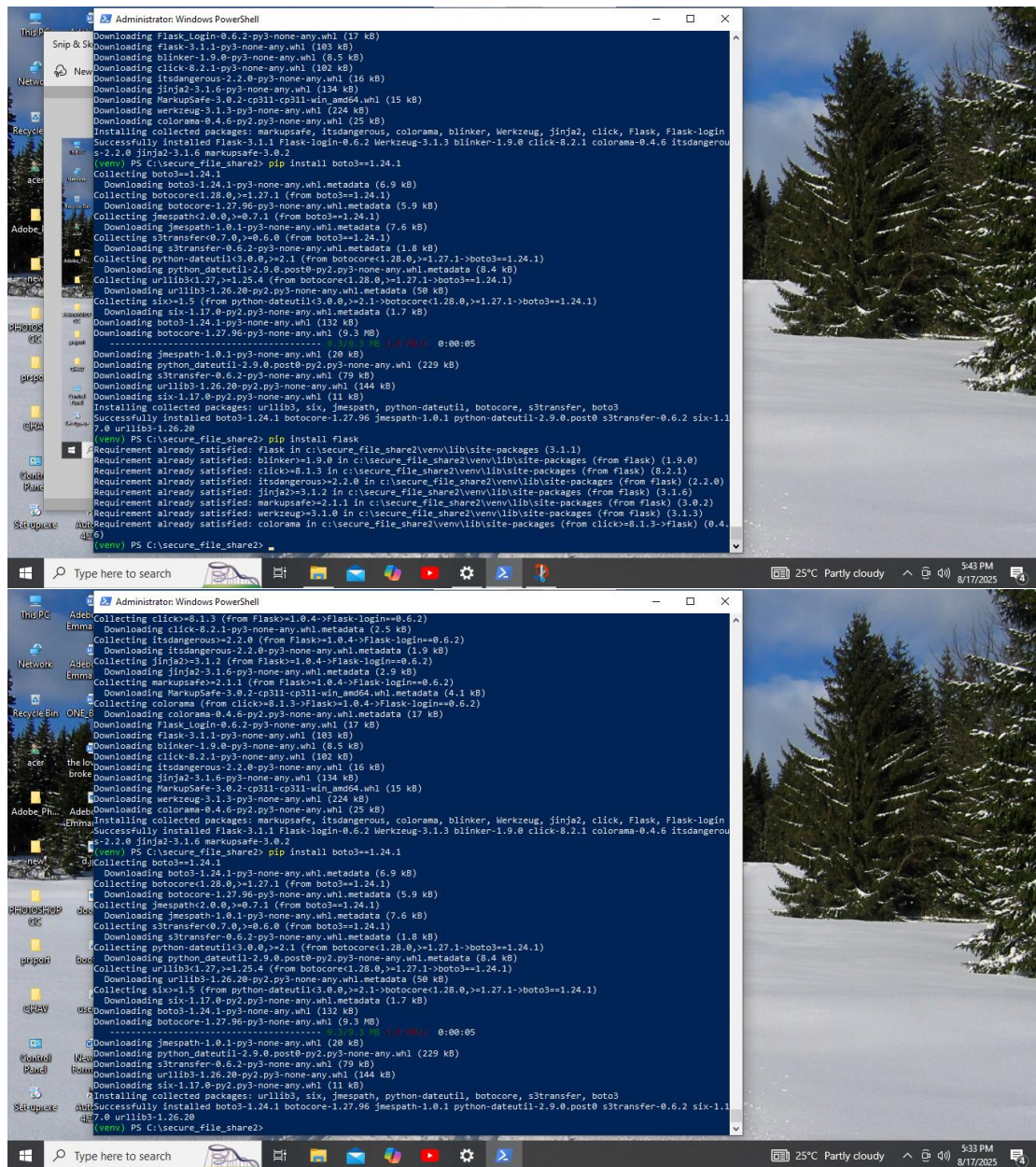
Page Showing the encrypted folder, test.txt etc.

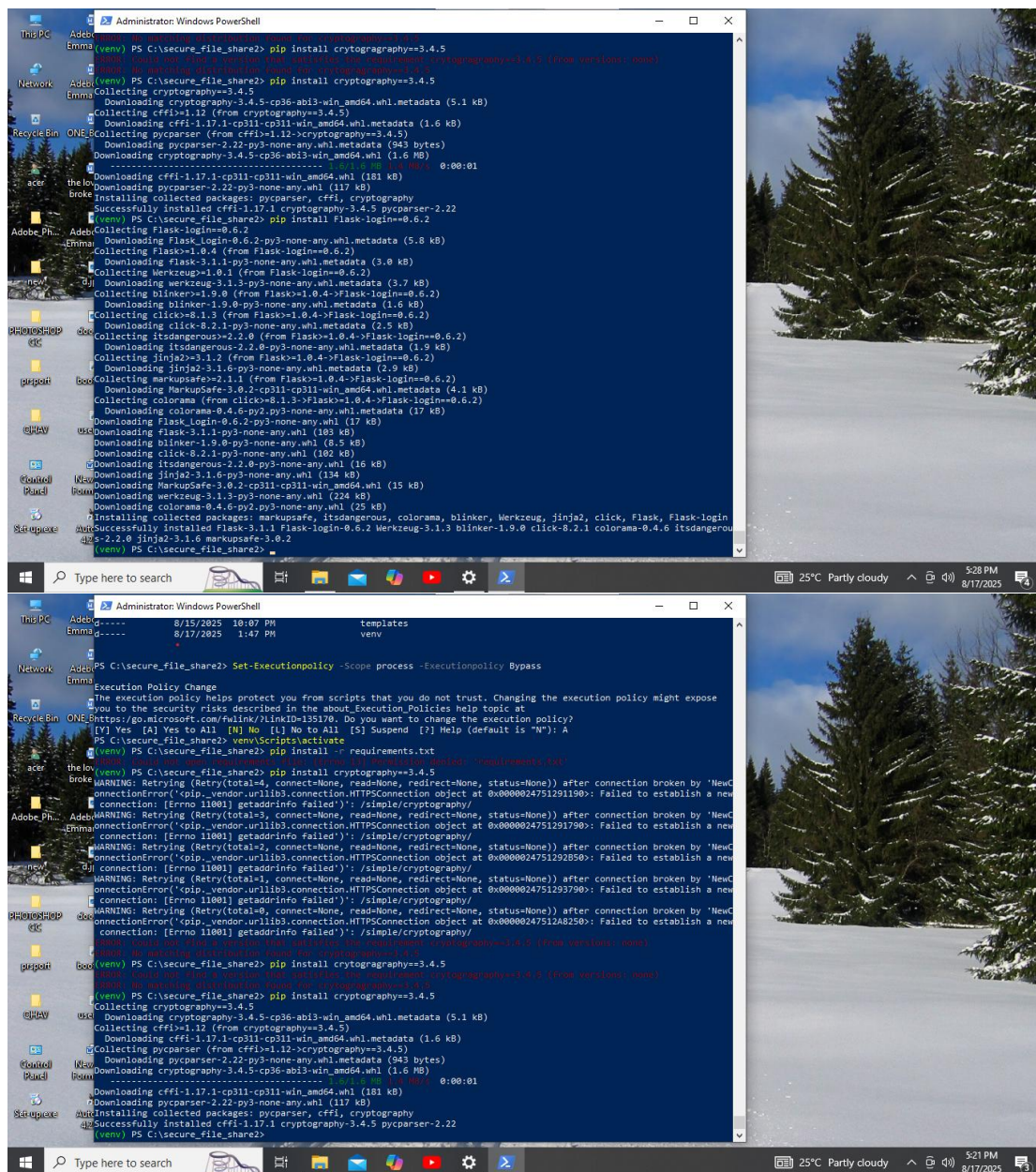


Encrypted file page









5. Challenges & Solutions

- Issue: Permission denied errors when installing requirements.
- Solution: Moved project folder to a different directory and ran commands with proper permissions.
- Issue: Misnaming of `__name__` variable in `app.py`
- Solution: Corrected the syntax to `__name__` which allowed Flask to run.
- Issue: Encrypted folder initially empty after upload.
- Solution: Properly clicked upload button, file succesfully appeared encrypted.

6. Conclusion & Lessons learned

Through this internship task, I gained hands-on experience in

- Setting up a python Flask project
- Implementing encryption with cryptography
- Managing file uploads and secure storage
- Troubleshooting python and environment setup issues
- Understanding real-world secure file transfer concepts

This project gave me practical exposure to Cybersecurity techniques and prepared me for working in secure software development.

7. References

- Flask Documentation: <https://flask.palletsprojects.com>
- Python Cryptography Library: <https://cryptography.io>
- Future interns internship Guide