



Report: Creating a Secure File Sharing Web Portal with AES Encryption on Windows

Author: *Bame Duncan Koko*

Date: *15 July 2025*

Technology Stack: Python 3, Flask, PyCharm Community
Edition, Windows CMD

Objective

To develop and document a secure file-sharing web portal that allows users to upload and download files with AES encryption implemented before storage and decryption on retrieval.

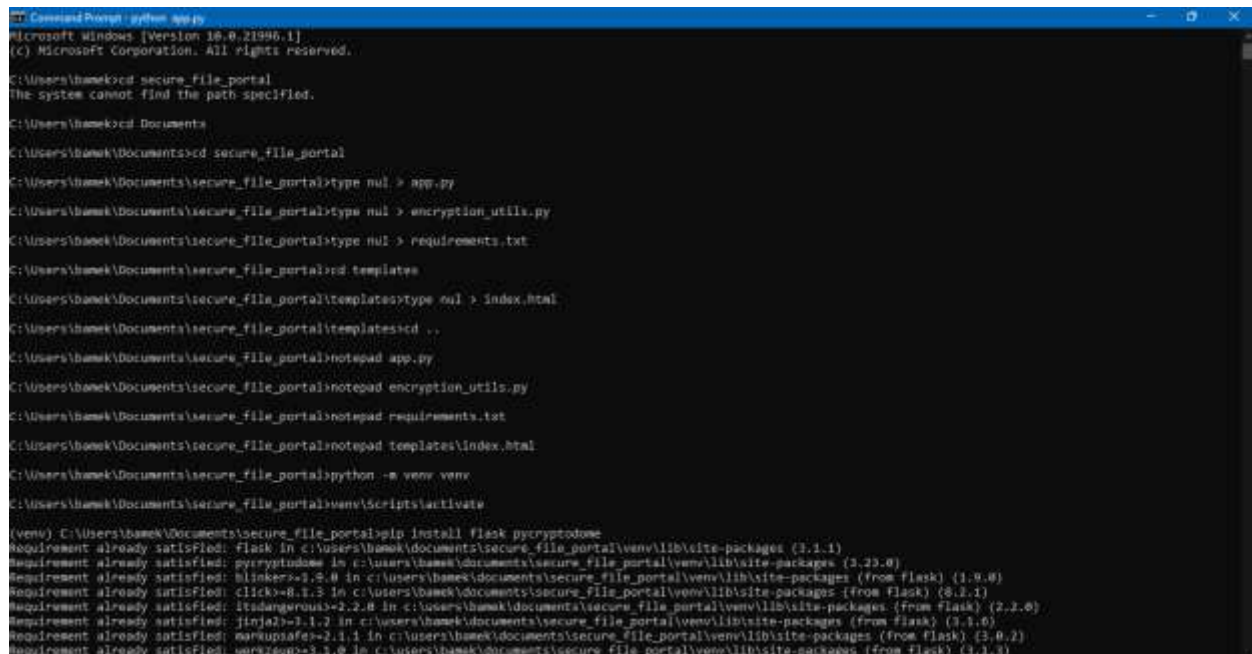
Tools and Technologies

- **Python 3.x** – Programming language
- **Flask** – Web framework
- **PyCryptodome** – AES encryption
- **Windows Command Prompt** – For project setup
- **PyCharm Community Edition** – For editing, running, and managing the project
- **Notepad (optional)** – For quick edits

Steps Taken

I installed Python 3 and added it to PATH, then installed PyCharm Community Edition for development. Using Command Prompt, I navigated to C:\Users\bamek\Documents, created a project folder named secure_file_portal, and entered it. I created and activated a virtual environment with python -m venv venv followed by venv\Scripts\activate. I then created a requirements.txt file containing flask and pycryptodome and installed these dependencies using pip install flask pycryptodome. Next, I built the project structure by creating the templates, uploads, and keys folders, and creating empty files app.py, encryption_utils.py, and requirements.txt, then within templates I created index.html. The final structure was:

C:\Users\bamek\Documents\secure_file_portal\ containing app.py, encryption_utils.py, requirements.txt, a templates folder with index.html, and uploads and keys directories.



```
Microsoft Windows [Version 10.0.21996.1]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bamek>cd secure_file_portal
The system cannot find the path specified.

C:\Users\bamek>cd Documents
C:\Users\bamek\Documents>cd secure_file_portal
C:\Users\bamek\Documents\secure_file_portal>type nul > app.py
C:\Users\bamek\Documents\secure_file_portal>type nul > encryption_utils.py
C:\Users\bamek\Documents\secure_file_portal>type nul > requirements.txt
C:\Users\bamek\Documents\secure_file_portal>rd templates
C:\Users\bamek\Documents\secure_file_portal\templates>type nul > index.html
C:\Users\bamek\Documents\secure_file_portal\templates>cd ..
C:\Users\bamek\Documents\secure_file_portal>notepad app.py
C:\Users\bamek\Documents\secure_file_portal>notepad encryption_utils.py
C:\Users\bamek\Documents\secure_file_portal>notepad requirements.txt
C:\Users\bamek\Documents\secure_file_portal>notepad templates\index.html
C:\Users\bamek\Documents\secure_file_portal>python -m venv venv
C:\Users\bamek\Documents\secure_file_portal>venv\scripts\activate

(venv) C:\Users\bamek\Documents\secure_file_portal>pip install flask pycryptodome
Requirement already satisfied: flask in c:\users\bamek\documents\secure_file_portal\venv\lib\site-packages (1.1.1)
Requirement already satisfied: pycryptodome in c:\users\bamek\documents\secure_file_portal\venv\lib\site-packages (3.20.0)
Requirement already satisfied: blinker>=1.5.0 in c:\users\bamek\documents\secure_file_portal\venv\lib\site-packages (from flask) (1.9.0)
Requirement already satisfied: click>=8.1.3 in c:\users\bamek\documents\secure_file_portal\venv\lib\site-packages (from flask) (8.2.1)
Requirement already satisfied: itsdangerous>=2.2.0 in c:\users\bamek\documents\secure_file_portal\venv\lib\site-packages (from flask) (2.2.0)
Requirement already satisfied: jinja2>=3.1.2 in c:\users\bamek\documents\secure_file_portal\venv\lib\site-packages (from flask) (3.1.6)
Requirement already satisfied: markupsafe>=2.1.1 in c:\users\bamek\documents\secure_file_portal\venv\lib\site-packages (from flask) (2.0.2)
Requirement already satisfied: werkzeug>=3.1.0 in c:\users\bamek\documents\secure_file_portal\venv\lib\site-packages (from flask) (3.1.0)
```

```

Command Prompt: python app.py
run      Run a development server.
shell    Run a shell in the app context.

(venv) C:\Users\bawek\Documents\secure_file_portal>pycryptodome
'pycryptodome' is not recognized as an internal or external command,
operable program or batch file.

(venv) C:\Users\bawek\Documents\secure_file_portal>
(venv) C:\Users\bawek\Documents\secure_file_portal>python app.py
Traceback (most recent call last):
  File "C:\Users\bawek\Documents\secure_file_portal\app.py", line 4, in <module>
    from encryption_utils import encrypt_file, decrypt_file
  File "C:\Users\bawek\Documents\secure_file_portal\encryption_utils.py", line 4, in <module>
    from encryption_utils import encrypt_file, decrypt_file
ImportError: cannot import name 'encrypt_file' from 'encryption_utils' (consider renaming 'C:\Users\bawek\Documents\secure_file_portal\encryption_utils.py' if it h
as the same name as a library you intended to import)

(venv) C:\Users\bawek\Documents\secure_file_portal>
(venv) C:\Users\bawek\Documents\secure_file_portal>pip install pycryptodome
Requirement already satisfied: pycryptodome in c:\users\bawek\documents\secure_file_portal\venv\lib\site-packages (3.23.0)

(venv) C:\Users\bawek\Documents\secure_file_portal>
(venv) C:\Users\bawek\Documents\secure_file_portal>pycryptodome
'pycryptodome' is not recognized as an internal or external command,
operable program or batch file.

(venv) C:\Users\bawek\Documents\secure_file_portal>(venv) C:\Users\bawek\Documents\secure_file_portal>python app.py
C:\Users\bawek\Documents\secure_file_portal was unexpected at this time.

(venv) C:\Users\bawek\Documents\secure_file_portal>python app.py
Traceback (most recent call last):
  File "C:\Users\bawek\Documents\secure_file_portal\app.py", line 4, in <module>
    from encryption_utils import encrypt_file, decrypt_file
  File "C:\Users\bawek\Documents\secure_file_portal\encryption_utils.py", line 4, in <module>
    from encryption_utils import encrypt_file, decrypt_file
ImportError: cannot import name 'encrypt_file' from 'encryption_utils' (consider renaming 'C:\Users\bawek\Documents\secure_file_portal\encryption_utils.py' if it h
as the same name as a library you intended to import)

```

```

(venv) C:\Users\bawek\Documents\secure_file_portal>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://127.0.0.1:5050
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PID: 283-408-307
127.0.0.1 - - [15/Jul/2025 22:43:25] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [15/Jul/2025 22:43:25] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [15/Jul/2025 22:43:36] "POST /upload HTTP/1.1" 302 -
127.0.0.1 - - [15/Jul/2025 22:43:36] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [15/Jul/2025 22:54:18] "GET /download/Databasell.png.enc HTTP/1.1" 200 -

```

Clipboard

Organize

New

Open

Select

←

→

⌵

⬆

📁

This PC > Documents > secure_file_portal >

★ Quick access

🖥 Desktop

⬆ Downloads

📄 Documents

📁 secure_file_portal

🖼 Pictures

📁 Documents

💿 HARD DRIVE (D:)

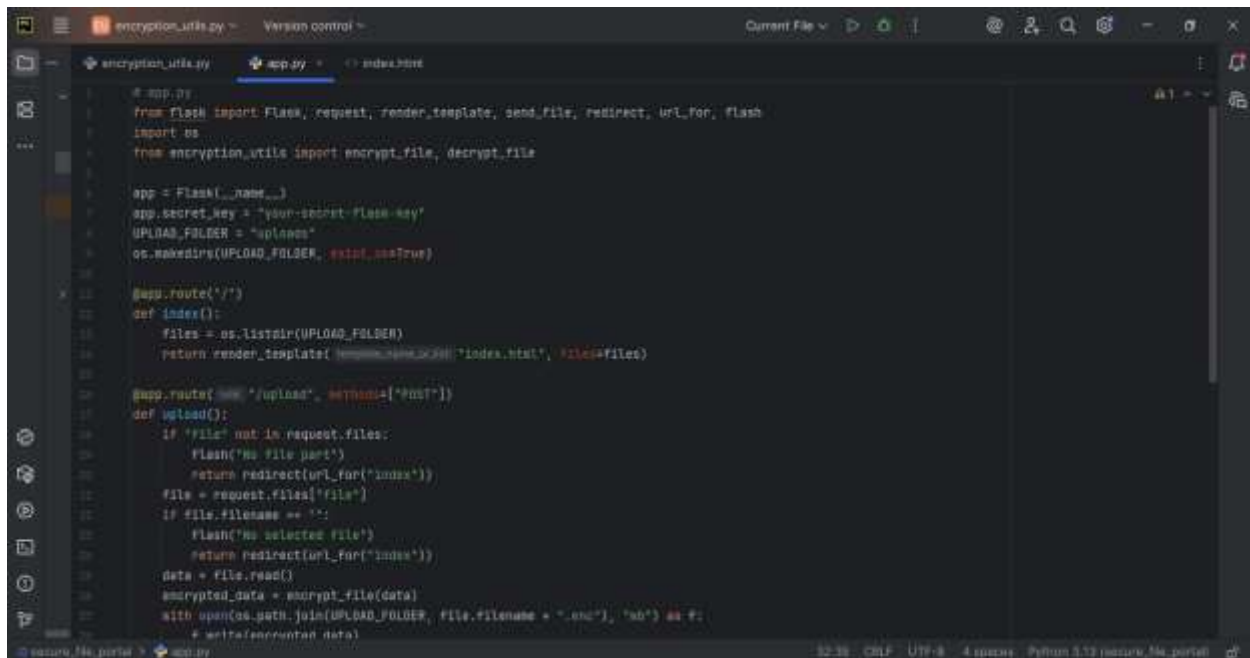
📁 templates

📁 VIDEO

Name	Date modified	Type	Size
📁 .idea	15/07/2025 22:41	File folder	
📁 __pycache__	15/07/2025 22:42	File folder	
📁 keys	15/07/2025 22:42	File folder	
📁 templates	15/07/2025 22:32	File folder	
📁 uploads	15/07/2025 22:58	File folder	
📁 venv	15/07/2025 22:08	File folder	
📄 app.py	15/07/2025 22:32	JetBrains PyChar...	2 KB
📄 encryption_utils.py	15/07/2025 22:41	JetBrains PyChar...	2 KB
📄 requirements.txt	15/07/2025 22:27	Text Document	0 KB

❏ Coding in PyCharm

1. Opened the folder in **PyCharm Community Edition**:
 - o File → Open → secure_file_portal
2. Edited app.py, encryption_utils.py, and templates/index.html with the provided code.
3. PyCharm's code highlighting helped ensure syntax correctness.

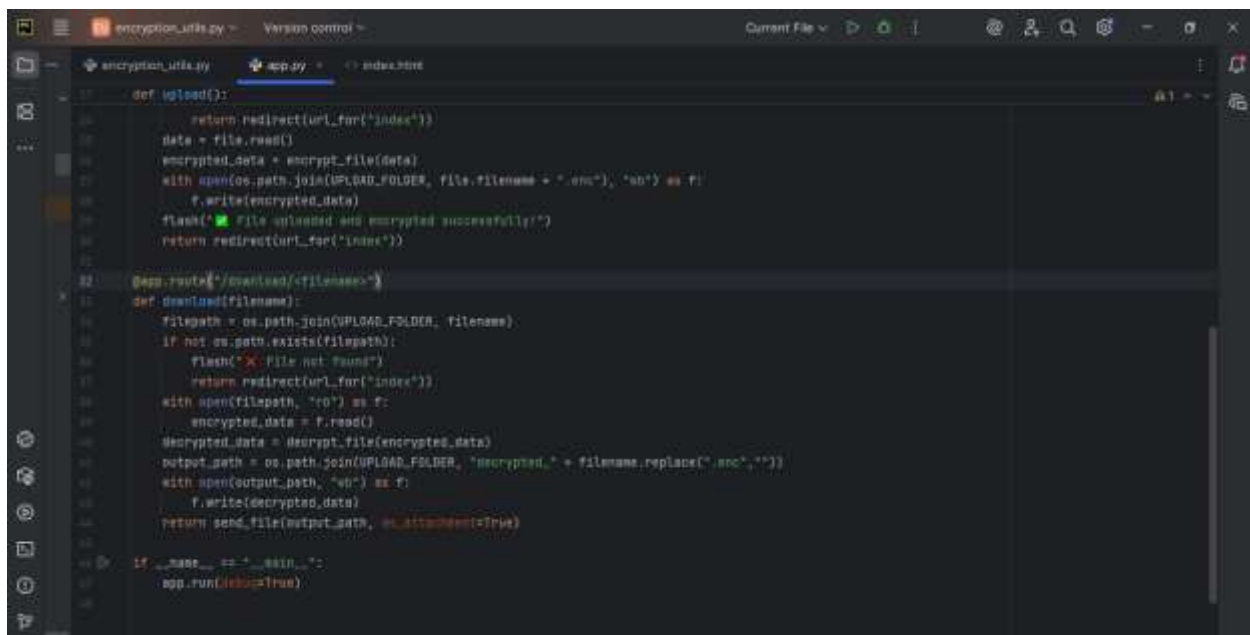


```
# app.py
from flask import Flask, request, render_template, send_file, redirect, url_for, flash
import os
from encryption_utils import encrypt_file, decrypt_file

app = Flask(__name__)
app.secret_key = "your-secret-flask-key"
UPLOAD_FOLDER = "uploads"
os.makedirs(UPLOAD_FOLDER, exist_ok=True)

@app.route("/")
def index():
    files = os.listdir(UPLOAD_FOLDER)
    return render_template("index.html", files=files)

@app.route("/upload", methods=['POST'])
def upload():
    if 'file' not in request.files:
        flash('No file part')
        return redirect(url_for("index"))
    file = request.files['file']
    if file.filename == '':
        flash('No selected file')
        return redirect(url_for("index"))
    data = file.read()
    encrypted_data = encrypt_file(data)
    with open(os.path.join(UPLOAD_FOLDER, file.filename + ".enc"), 'wb') as f:
        f.write(encrypted_data)
```



```
def upload():
    return redirect(url_for("index"))
    data = file.read()
    encrypted_data = encrypt_file(data)
    with open(os.path.join(UPLOAD_FOLDER, file.filename + ".enc"), 'wb') as f:
        f.write(encrypted_data)
    flash("File uploaded and encrypted successfully!")
    return redirect(url_for("index"))

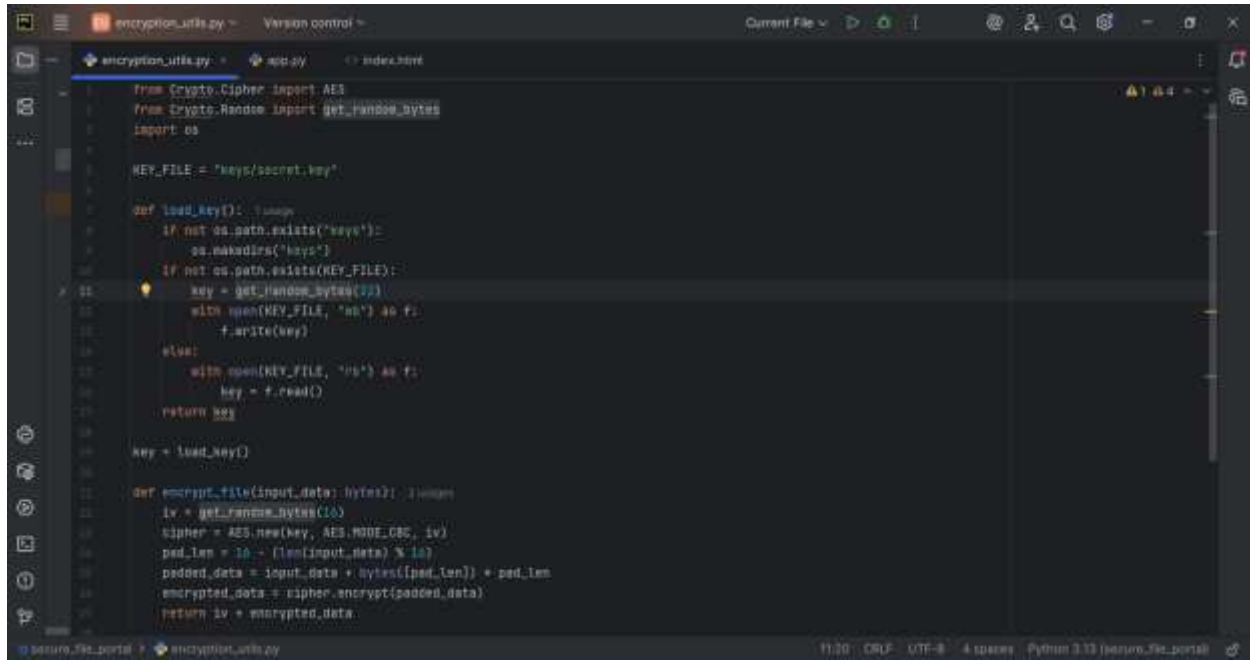
@app.route("/download/<filename>")
def download(filename):
    filepath = os.path.join(UPLOAD_FOLDER, filename)
    if not os.path.exists(filepath):
        flash("File not found")
        return redirect(url_for("index"))
    with open(filepath, "rb") as f:
        encrypted_data = f.read()
    decrypted_data = decrypt_file(encrypted_data)
    output_path = os.path.join(UPLOAD_FOLDER, "decrypted_" + filename.replace(".enc", ""))
    with open(output_path, "wb") as f:
        f.write(decrypted_data)
    return send_file(output_path, as_attachment=True)

if __name__ == "__main__":
    app.run(debug=True)
```

Implementing AES Encryption

In encryption_utils.py:

- Loaded/stored 256-bit key in keys/secret.key.
- Created encrypt_file() and decrypt_file() functions using AES-256 CBC.



```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
import os

KEY_FILE = 'keys/secret.key'

def load_key():
    if not os.path.exists('keys'):
        os.makedirs('keys')
    if not os.path.exists(KEY_FILE):
        key = get_random_bytes(32)
        with open(KEY_FILE, 'wb') as f:
            f.write(key)
    else:
        with open(KEY_FILE, 'rb') as f:
            key = f.read()
    return key

key = load_key()

def encrypt_file(input_data: bytes) -> bytes:
    iv = get_random_bytes(16)
    cipher = AES.new(key, AES.MODE_CBC, iv)
    pad_len = 16 - (len(input_data) % 16)
    padded_data = input_data + bytes([pad_len]) * pad_len
    encrypted_data = cipher.encrypt(padded_data)
    return iv + encrypted_data
```

```
def load_key():
    with open(KEY_FILE, 'rb') as f:
        key = f.read()
    return key

key = load_key()

def encrypt_file(input_data: bytes):
    iv = get_random_bytes(16)
    cipher = AES.new(key, AES.MODE_CBC, iv)
    pad_len = 16 - (len(input_data) % 16)
    padded_data = input_data + bytes([pad_len]) * pad_len
    encrypted_data = cipher.encrypt(padded_data)
    return iv + encrypted_data

def decrypt_file(encrypted_data: bytes):
    iv = encrypted_data[:16]
    data = encrypted_data[16:]
    cipher = AES.new(key, AES.MODE_CBC, iv)
    decrypted_padded = cipher.decrypt(data)
    pad_len = decrypted_padded[-1]
    return decrypted_padded[:pad_len]
```

User Interface

Created a minimal HTML page in templates/index.html with:

- File upload form
- Links to download encrypted files

```
<!DOCTYPE html>
<html>
<head>
<title>Secure File Portal</title>
</head>
<body>
<h1>Secure File Sharings</h1>

<h2>Upload a File</h2>
<form action="/upload" method="post" enctype="multipart/form-data">
<input type="file" name="file"/>
<button type="submit">Upload</button>
</form>

<h2>Available Files</h2>
<ul>
<% for f in files %>
<li>{{ f }} - <a href="/download/{{f}}">Download & Decrypt</a></li>
<% endfor %>
</ul>

<% with messages = get_flashed_messages() %>
<% if messages %>
<ul>
<% for msg in messages %>
<li>{{ msg }}</li>
<% endfor %>
</ul>
</if>
```

► Running the Application

In PyCharm:

- Right-clicked app.py → **Run 'app'**
or via Command Prompt:

cmd
Copy code
python app.py

Output:

csharp
Copy code
* Running on http://127.0.0.1:5000/

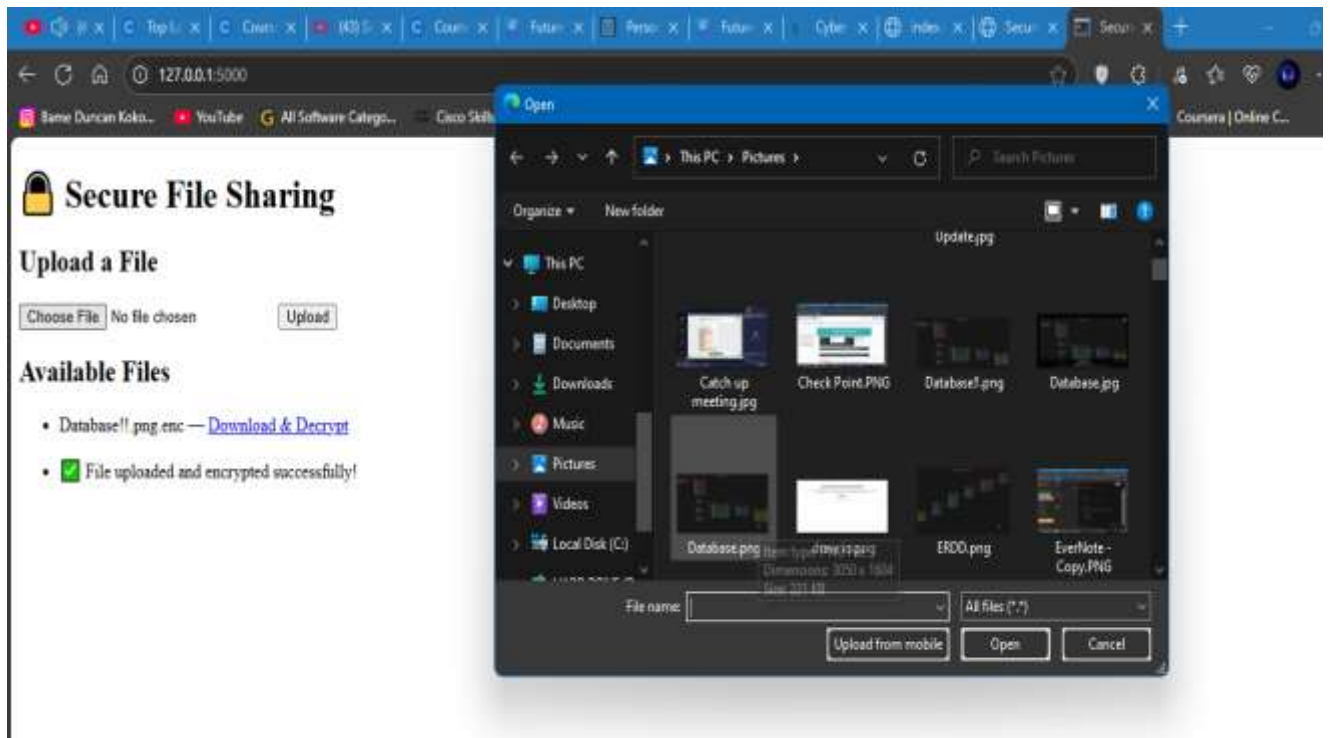
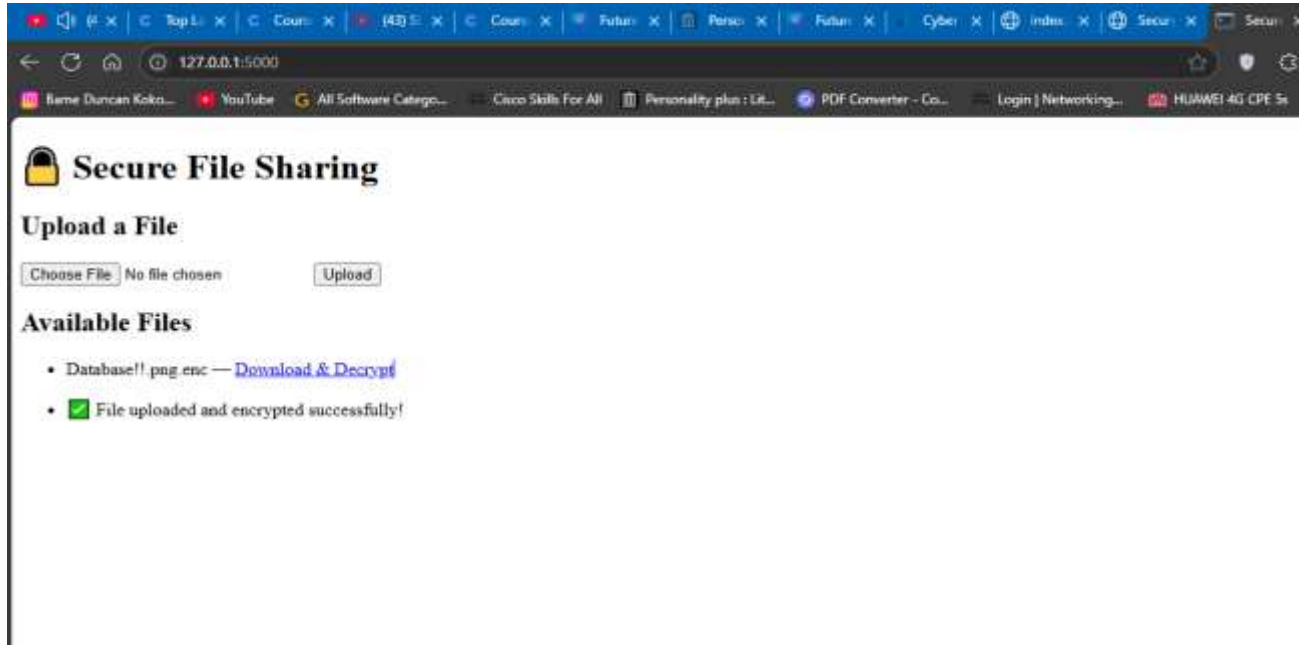
Accessed in browser: <http://127.0.0.1:5000>

```
(venv) C:\Users\bamek\Documents\secure_file_portal>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: 283-498-297
127.0.0.1 - - [15/Jul/2025 22:43:25] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [15/Jul/2025 22:43:25] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [15/Jul/2025 22:43:36] "POST /upload HTTP/1.1" 302 -
127.0.0.1 - - [15/Jul/2025 22:43:36] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [15/Jul/2025 22:58:18] "GET /download/Database!!.png.enc HTTP/1.1" 200 -
127.0.0.1 - - [15/Jul/2025 23:08:14] "POST /upload HTTP/1.1" 302 -
127.0.0.1 - - [15/Jul/2025 23:08:14] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [15/Jul/2025 23:09:27] "GET /download/Database!!.png.enc HTTP/1.1" 200 -
```

```
(venv) C:\Users\bamek\Documents\secure_file_portal>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: 283-498-297
127.0.0.1 - - [15/Jul/2025 22:43:25] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [15/Jul/2025 22:43:25] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [15/Jul/2025 22:43:36] "POST /upload HTTP/1.1" 302 -
127.0.0.1 - - [15/Jul/2025 22:43:36] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [15/Jul/2025 22:58:18] "GET /download/Database!!.png.enc HTTP/1.1" 200 -
127.0.0.1 - - [15/Jul/2025 23:08:14] "POST /upload HTTP/1.1" 302 -
127.0.0.1 - - [15/Jul/2025 23:08:14] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [15/Jul/2025 23:09:27] "GET /download/Database!!.png.enc HTTP/1.1" 200 -
```


✓ Testing

- Uploaded a sample file, confirmed .enc file created.
- Downloaded the file, verified it decrypted correctly.



Top L... x | C Course x | (43) Si x | C Course x | F

127.0.0.1:5000

Bame Duncan Koko... YouTube All Software Catego... Cisco Skills For All

Secure File Sharing

Upload a File

Database.png

Available Files

- Database!! .png .enc — [Download & Decrypt](#)
- ☒ File uploaded and encrypted successfully!

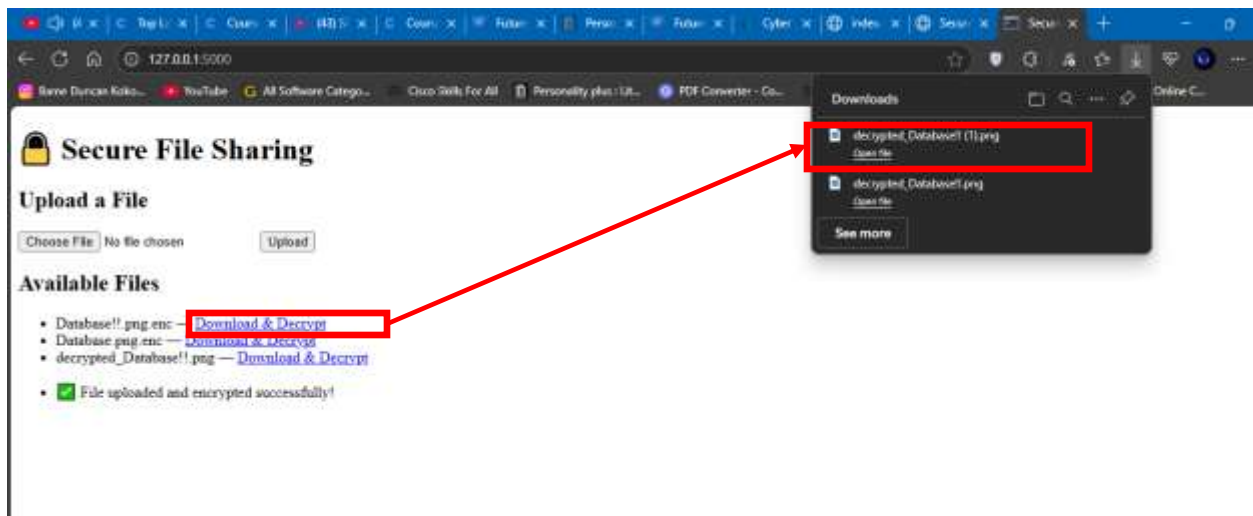
Secure File Sharing

Upload a File

No file chosen

Available Files

- Database!! .png .enc — [Download & Decrypt](#)
- Database.png .enc — [Download & Decrypt](#)
- decrypted_Database!! .png — [Download & Decrypt](#)
- ☒ File uploaded and encrypted successfully!



Security Measures

- AES-256 CBC encryption implemented.
- Key stored in keys/secret.key.
- Encrypted files stored in uploads.
- Flask secret_key set for sessions.
- Recommendation: use HTTPS and stronger key management in production.

Conclusion

The secure file sharing portal was successfully developed and run locally on Windows using **PyCharm Community Edition**.

✓ Next Steps / Future Improvements:

- Add authentication and access control.
- Deploy to a cloud server with HTTPS.
- Use a secrets manager for key storage.