

# File Security System – Security Overview Document

## Project Overview

The **File Security System** is a web-based application designed to securely handle file storage, upload, and download using **encryption**. The system ensures confidentiality, integrity, and controlled access to sensitive files.

---

## Objectives

- Protect files from unauthorized access
  - Enable secure upload and download functionality
  - Implement encryption for stored files
  - Demonstrate cybersecurity best practices in web applications
- 

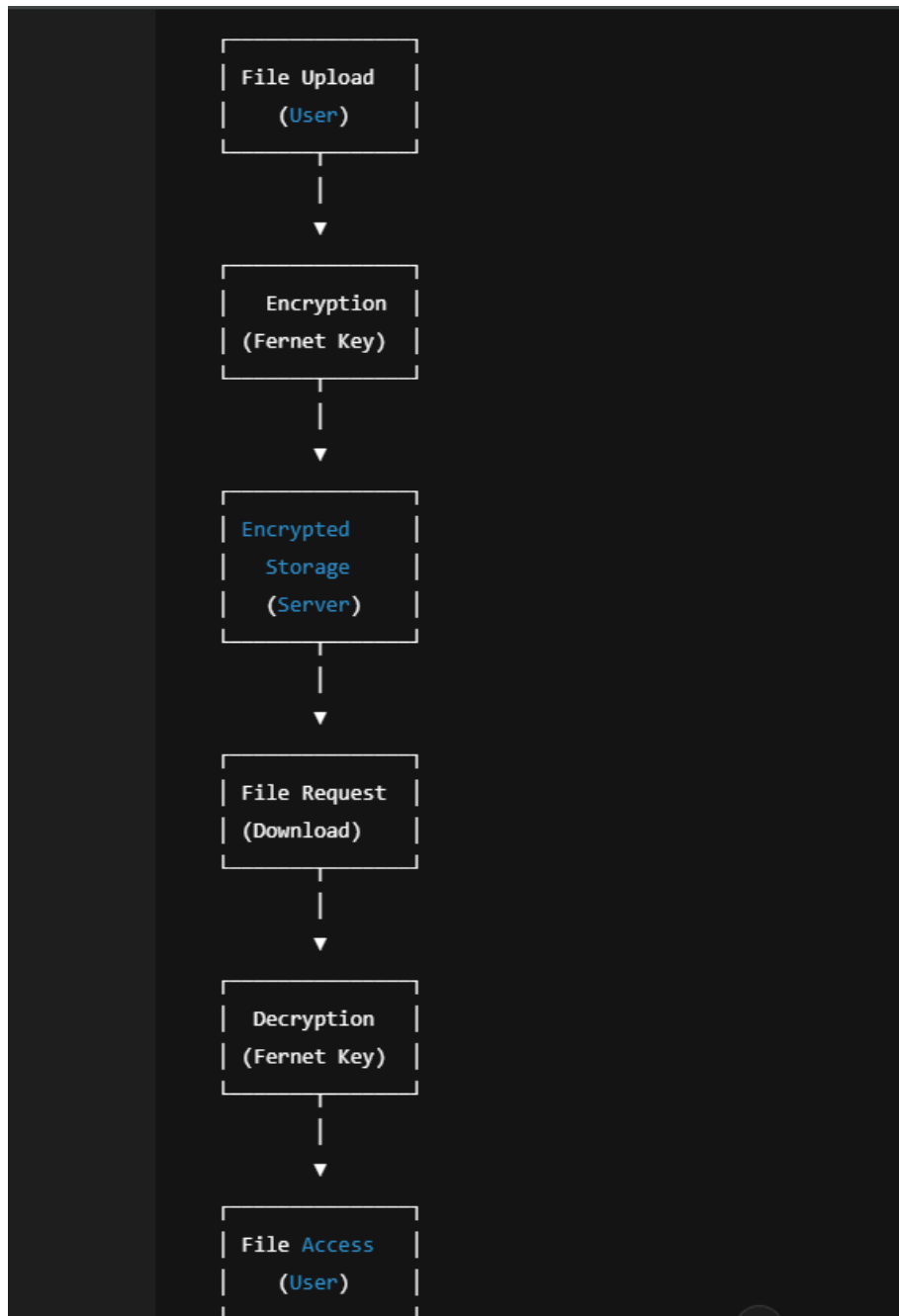
## Tools & Technologies Used

- **Backend:** Python (Flask Framework)
  - **Frontend:** HTML, Tailwind CSS
  - **Encryption:** Python Cryptography (Fernet)
  - **IDE:** Visual Studio Code
- 

## System Architecture / Website Flow

1. **File Upload** – User uploads a file through the web interface
2. **Encryption** – The system encrypts the file using **Fernet key** before saving
3. **Storage** – Encrypted files are stored securely in the server directory

4. **File Download** – When requested, the file is **decrypted** and shared with the authenticated user
5. **Key Management** – Encryption keys are generated, stored, and reused for consistent encryption/decryption





## File Security Tool

Choose what you want to do with your files

Encrypt a File

Decrypt a File

Files are processed securely and never stored permanently.

2025 | Developed by [Pratik Sayankar](#)



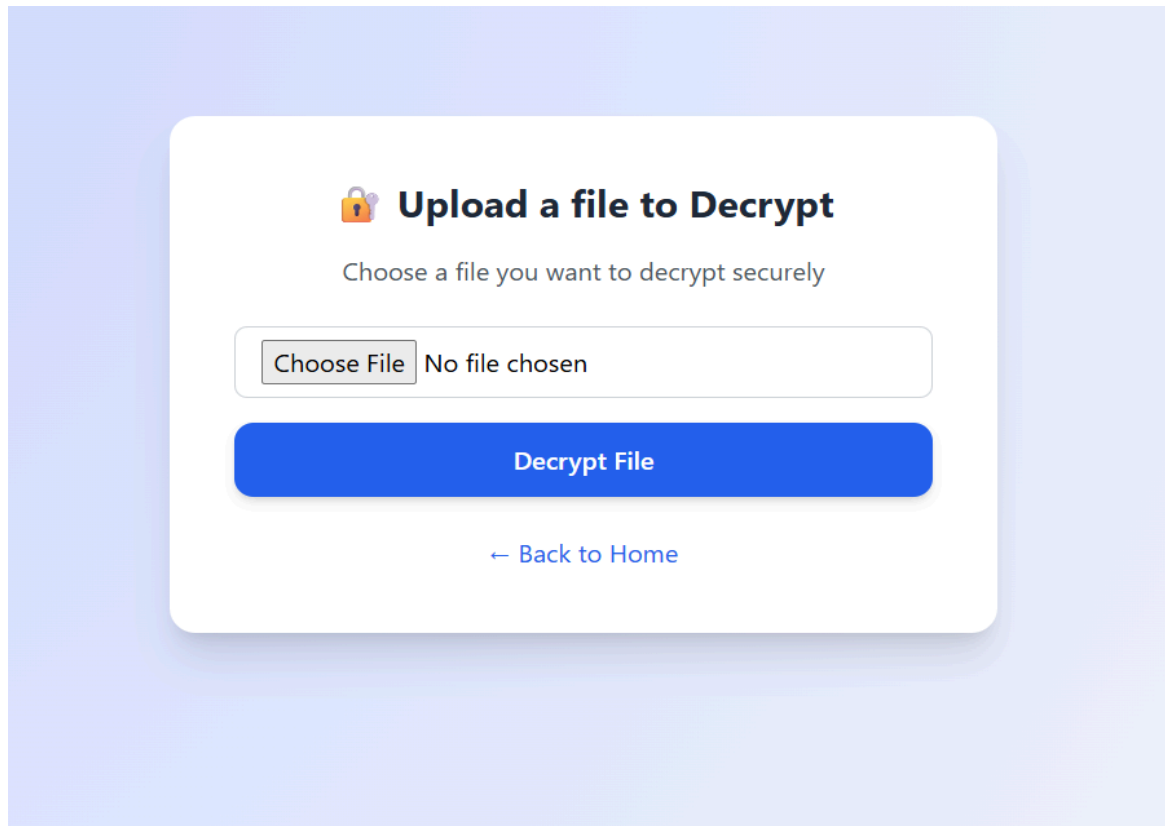
## Upload a file to Encrypt

Choose a file you want to encrypt securely

Choose File No file chosen

Encrypt File

[← Back to Home](#)



## Key Features

- Secure **file upload & download**
- Strong **encryption (AES via Fernet)**
- Protection against unauthorized access
- Lightweight **Flask web interface**

### /project-root

```
| — main.py      # Flask application
| — secret.key   # Encryption key file
| — /uploads     # Encrypted file storage
| — /templates   # HTML templates
|   | — index.html
|   | — preview.html
|   | — success.html
```

## Main Code Snippets:

- Key Generation & Loading
- Encryption & Decryption Functions
- Upload & Download Routes (Flask)
- HTML Frontend Templates

### – [main.py](#)

```
FUTURE_CS_03 > main.py > ...
1  from flask import Flask, render_template, request, send_file, redirect, url_for
2  from cryptography.fernet import Fernet
3  import os
4
5  app = Flask(__name__)
6  UPLOAD_FOLDER = 'uploads'
7  os.makedirs(UPLOAD_FOLDER, exist_ok=True)
8
9  # Load or generate a persistent encryption key
10 KEY_FILE = 'secret.key'
11 if os.path.exists(KEY_FILE):
12     with open(KEY_FILE, 'rb') as f:
13         key = f.read()
14 else:
15     key = Fernet.generate_key()
16     with open(KEY_FILE, 'wb') as f:
17         f.write(key)
18
19 fernet = Fernet(key)
20
21 @app.route('/')
22 def index():
23     return render_template('index.html')
24
25 @app.route('/encrypt_upload', methods=['GET', 'POST'])
26 def encrypt_upload():
27     if request.method == 'POST':
28         file = request.files['file']
29         if file:
30             filepath = os.path.join(UPLOAD_FOLDER, file.filename)
31             file.save(filepath)
32
33             with open(filepath, 'rb') as f:
34                 encrypted_data = fernet.encrypt(f.read())
35
36             encrypted_path = os.path.join(UPLOAD_FOLDER, 'encrypted_' + file.filename)
37             with open(encrypted_path, 'wb') as f:
38                 f.write(encrypted_data)
39
40             return render_template('success.html', action='encrypted', filename='encrypted_' + file.filename)
41         return render_template('preview.html', action='encrypt')
42
43 @app.route('/decrypt_upload', methods=['GET', 'POST'])
44 def decrypt_upload():
45     if request.method == 'POST':
46         file = request.files['file']
47         if file:
48             filepath = os.path.join(UPLOAD_FOLDER, file.filename)
49             file.save(filepath)
```

```

50
51         try:
52             with open(filepath, 'rb') as f:
53                 decrypted_data = fernet.decrypt(f.read())
54
55             decrypted_path = os.path.join(UPLOAD_FOLDER, 'decrypted_' + file.filename)
56             with open(decrypted_path, 'wb') as f:
57                 f.write(decrypted_data)
58
59             return render_template('success.html', action='decrypted', filename='decrypted_' + file.filename)
60         except Exception as e:
61             return f"<h2>Error during decryption: {str(e)}</h2>"
62     return render_template('preview.html', action='decrypt')
63
64 @app.route('/download/<filename>')
65 def download(filename):
66     path = os.path.join(UPLOAD_FOLDER, filename)
67     return send_file(path, as_attachment=True)
68
69 if __name__ == '__main__':
70     app.run(debug=True)
71

```

## – index.html

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>File Security Tool</title>
6      <script src="https://cdn.tailwindcss.com"></script>
7  </head>
8  <body class="min-h-screen flex items-center justify-center bg-gray-100 bg-[radial-gradient(circle_at_top_left,_#c7d2fe,_#e0e7ff,_#f9fafb)]">
9
10
11      <div class="bg-white shadow-xl rounded-2xl p-10 max-w-lg w-full text-center">
12          <h1 class="text-3xl font-bold text-gray-800 mb-4">🔒 File Security Tool</h1>
13          <p class="text-gray-600 mb-8">Choose what you want to do with your files</p>
14
15          <div class="grid gap-6">
16              <!-- Encrypt Button -->
17              <a href="/encrypt_upload"
18                  class="flex items-center justify-center gap-3 bg-blue-600 hover:bg-blue-700 text-white font-semibold py-3 rounded-xl transition transform hover:scale-105 shadow-md">
19                  <span>Encrypt a File</span>
20              </a>
21
22              <!-- Decrypt Button -->
23              <a href="/decrypt_upload"
24                  class="flex items-center justify-center gap-3 bg-green-600 hover:bg-green-700 text-white font-semibold py-3 rounded-xl transition transform hover:scale-105 shadow-md">
25                  <span>Decrypt a File</span>
26              </a>
27          </div>
28
29          <p class="text-xs text-gray-400 mt-8">
30              Files are processed securely and never stored permanently.
31          </p>
32      </div>
33
34      <div class="text-center mt-6" align="center" style="position: fixed; bottom: 30px;">
35          <!-- Footer -->
36          <footer class="mt-10 text-center text-gray-500 text-sm">
37              <h3 class="font-medium">
38                  2025 | Developed by
39                  <a href="https://www.linkedin.com/in/pratik-sayankar-806355200/" target="_blank" class="text-blue-600">Pratik Sayankar</a>
40              </h3>
41          </footer>
42      </div>
43
44
45 </body>
46 </html>

```

## – preview.html

```
FUTURE_CS_03 > templates > preview.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>{{ action|capitalize }} File</title>
6    <script src="https://cdn.tailwindcss.com"></script>
7  </head>
8  <body class="min-h-screen flex items-center justify-center bg-gray-100
9    bg-[radial-gradient(circle_at_top_left,_#c7d2fe,_#e0e7ff,_#f9fafb)]">
10
11    <div class="bg-white shadow-xl rounded-2xl p-10 max-w-lg w-full text-center">
12      <h2 class="text-2xl font-bold text-gray-800 mb-4">📁 Upload a file to {{ action|capitalize }}</h2>
13      <p class="text-gray-600 mb-6">Choose a file you want to {{ action }} securely</p>
14
15      <form method="POST" enctype="multipart/form-data" class="space-y-4">
16        <input type="file" name="file" required
17          class="w-full border border-gray-300 rounded-lg px-4 py-2 focus:ring focus:ring-blue-300">
18
19        <button type="submit"
20          class="w-full bg-blue-600 hover:bg-blue-700 text-white font-semibold py-3 rounded-xl transition transform hover:scale-105 shadow-md">
21          {{ action|capitalize }} File
22        </button>
23      </form>
24
25      <a href="/" class="inline-block mt-6 text-blue-600 hover:underline">+ Back to Home</a>
26    </div>
27  </body>
28 </html>
29
30
```

## – success.html

```
FUTURE_CS_03 > templates > success.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>Success</title>
6    <script src="https://cdn.tailwindcss.com"></script>
7  </head>
8  <body class="min-h-screen flex items-center justify-center bg-gray-100
9    bg-[radial-gradient(circle_at_top_left,_#c7d2fe,_#e0e7ff,_#f9fafb)]">
10
11    <div class="bg-white shadow-xl rounded-2xl p-10 max-w-lg w-full text-center">
12      <h2 class="text-2xl font-bold text-green-600 mb-4">✅ File {{ action|capitalize }} Successfully!</h2>
13      <p class="text-gray-700 mb-6"><strong>{{ filename }}</strong></p>
14
15      <div class="flex flex-col gap-4">
16        <a href="/download/{{ filename }}"
17          class="bg-green-600 hover:bg-green-700 text-white font-semibold py-3 rounded-xl transition transform hover:scale-105 shadow-md">
18          ↓ Download File
19        </a>
20        <a href="/"
21          class="bg-gray-600 hover:bg-gray-700 text-white font-semibold py-3 rounded-xl transition transform hover:scale-105 shadow-md">
22          🏠 Back to Home
23        </a>
24      </div>
25    </div>
26  </body>
27 </html>
28
```

## Security Considerations

- Files are encrypted before storage
- Encryption key is kept persistent and safe

---

## Learning Outcomes

- Applied cryptography in real-world web apps
- Understood file security workflow
- Improved problem-solving in secure coding
- Learned integration of backend + frontend + security
- Strengthened fundamentals of confidentiality, integrity, availability (CIA triad)

---

## Acknowledgement

This project was developed as part of the Future Intern Program.

Special thanks to Future Intern for the opportunity to enhance my cybersecurity and software development skills.