🔐 Security Overview Document

"Security is not a feature – it's a mindset."

Project: Secure File Sharing System
Developer: Saumyata Nepal
Internship Program: Future Interns – Cyber Security Task 3
Date: 20 July 2025

---

1. Encryption Used: AES (Advanced Encryption Standard)

- AES is a **symmetric encryption algorithm** used to secure data by converting plaintext into ciphertext using a secret key.
- I used **AES-256-CBC mode** (256-bit key size, Cipher Block Chaining) via the PyCryptodome library in Python.
- AES is **widely trusted** and used in banking, military, and enterprise systems for protecting sensitive data.

---

2. Key Management

- A **randomly generated secret key** (32 bytes for AES-256) is used for encryption and decryption.
- The key is currently **hardcoded or stored in a config variable** during development.
- For real-world usage, the key should be securely stored using:
  - Environment variables
  - Key Management Services (e.g., AWS KMS, Azure Key Vault)
  - .env files (with .gitignore to prevent exposure in version control)

---

3. File Handling Process

- Users upload a file via the frontend.

- The file is **encrypted using AES** and stored in the uploads/ directory with a .enc extension.
- To download, users enter the encrypted filename, and the system **decrypts it in real-time**, allowing secure file download.

---

4. Security Considerations

- Files are **never stored in plain text** on the server.
- The encryption key is **not exposed to users** or stored alongside the file.
- Only encrypted files are kept in storage, reducing data breach impact.
- HTTPS is recommended for deployment to secure file transfers.

---

5. Future Improvements

- Implement a **secure login/authentication system** for users.
- Add **file size limits** and type restrictions to prevent abuse.
- Store encryption keys in a **dedicated key vault** instead of local memory.
- Enable file **expiration or auto-deletion** after a set time for added security.

Summary

This project demonstrates how encryption, key management, and secure file storage can be integrated into a web app. Using AES ensures **confidentiality**, while Flask provides a lightweight platform for secure file sharing.