**FUTURE INTERNS TASK\_2(CYBER SECURITY)**

Name : Spoorthi K.G

Domain: Cyber Security

Title: Secure File Sharing System

**🚀 Project Vision**

Empower users to protect their files through a simple, transparent, and ethically secure web interface. This application allows users to upload files and perform encryption or decryption using symmetric cryptography—prioritizing trust, usability, and data protection.

**🧠 Design Principles**

* **Transparency First**: No hidden logic or silent data storage.
* **User Control**: Encryption is intentional, not automatic.
* **Calm Simplicity**: Minimalist interface with intuitive navigation.

**🛠️ Technology Stack**

The backend is powered by Flask, a lightweight Python framework that handles routing and file operations. Encryption is implemented using the cryptography.fernet library, which provides AES encryption in CBC mode with HMAC for integrity. The frontend uses basic HTML and inline CSS for simplicity and accessibility. Files are stored locally in an uploads/ directory, and the encryption key is saved persistently in a secret.key file to allow repeatable decryption.

**🧱 Application Structure**

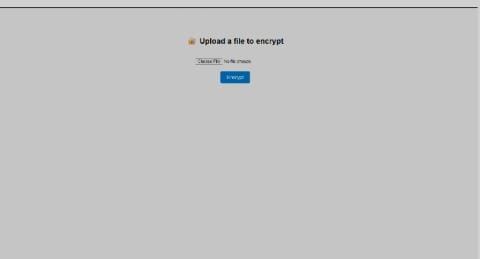
The core logic resides in main.py, which manages routing and cryptographic operations. HTML templates are stored in the templates/ folder and include three key pages: index.html, preview.html, and success.html. Uploaded files are saved in the uploads/ directory, and the Fernet key is stored in secret.key to ensure consistent encryption and decryption across sessions.

**🖥️ User Interface Flow**

* The user journey begins at the home screen (index.html), which features two clear action buttons: Encrypt and Decrypt. The design is minimalist, using light colors and friendly icons to enhance accessibility.



* Upon selecting an action, users are directed to the upload screen (preview.html). This page presents a file upload form and dynamically adjusts its text and submit button based on the chosen operation—either encryption or decryption.



* After the file is processed, users land on the confirmation screen (success.html). This page displays a success message, provides a direct download link for the processed file, and includes a navigation button to return to the home page.



**🔐Encryption Strategy**

The application uses Fernet encryption, which combines AES in CBC mode with HMAC for integrity verification. A single Fernet key is generated and stored in secret.key, allowing secure and repeatable decryption without requiring users to manage keys manually. This approach ensures strong confidentiality and integrity while keeping the implementation lightweight and user-friendly.

**🔄 Operational Workflow**

1. The user visits the home page and selects either Encrypt or Decrypt.
2. They are redirected to the preview page, where they upload a file.
3. The backend processes the file using Fernet encryption or decryption.
4. The success page confirms the operation and provides a download link.