Security Overview – Secure File Portal

Project: Secure File Upload/Download Portal
Platform: Python Flask
Encryption: AES-CBC
Prepared for: Internship Task 3

---

1. Encryption

- AES-CBC (Cipher Block Chaining) for all files.
- 16-byte AES key stored in environment variable (.env).
- Each file uses a **unique random IV** for enhanced security.
- Files are encrypted **before saving**; no plaintext storage.

2. Passwords

- Admin password hashed using **bcrypt**.
- Passwords are never stored or transmitted in plaintext.
- Authentication required to access upload/download routes.

3. File Handling

- Temporary files exist **only during encryption/decryption** and are deleted immediately.
- Each user has a **separate folder** for file isolation.
- Supports **multiple file uploads** with real-time progress display.
- Downloads are **decrypted on-the-fly**.

4. Key Management

- AES key stored securely in **environment variables (.env)**.
- Key is never hardcoded or committed to version control.
- Only the server uses the key for encryption/decryption.

5. Session & Authentication

- Flask sessions maintain logged-in users.
- Upload/download routes protected by authentication.
- Logout destroys the session.

6. Additional Security Measures

- Temporary decrypted files automatically deleted.
- Per-user folder isolation prevents unauthorized access.
- Secure handling during file upload/download ensures data integrity.
- Environment variables protect sensitive information.

Prepared by: *Shahana Parveen*