**TASK 3: Security Overview Report — Secure File Sharing System**
**Student Name:** Harsha A
**Project Title:** Secure File Sharing System
**Task Number:** 3
**Date:** 24 July 2025

---

**Objective:**

To analyze the secure file sharing system's current security posture and provide a concise report on potential vulnerabilities, implemented protections, and suggested improvements.

---

**Key Observations:**

**1. File Encryption**

- Files are encrypted using AES-128 encryption with a pre-shared key.
- Key example: AES_KEY=secretkey1234567
- **Strength:** Provides confidentiality for uploaded files.
- **Risk:** Hardcoding the key in code is insecure — it should be stored in environment variables.

**2. Upload/Download Features**

- Encrypted files can be uploaded and downloaded by the user.
- Access control is role-based (e.g., user vs. admin).

**3. Authentication & Access**

- Basic authentication is in place.
- **Strength:** Separates user and admin functionalities.
- **Risk:** No two-factor authentication; session tokens not visible — may need session timeout or refresh token checks.

**4. Security Issues Noted**

- Some command-line inputs such as AES_KEY=secretkey1234567 returned errors.
- File decryption issues or missing files (e.g., index.html.enc) noted.
- Possible improper error handling or missing decryption key.

---

**Recommendations:**

- **Secure AES Key Storage:** Store the AES key in a .env file or a secure key vault, not directly in the code.
- **Error Handling:** Add clear alerts for file not found, decryption failure, or invalid uploads.

- **Audit Logging:** Implement logs for user actions, file access, and admin changes.
- **Improve Authentication:** Use token-based login or OAuth and optionally 2FA.
- **Input Validation:** Ensure filenames, keys, and uploads are strictly validated to prevent injection or traversal attacks.

---

**Conclusion:**

The Secure File Sharing System demonstrates basic functionality in encryption and role-based file access. However, improvements in secure key management, authentication, error handling, and logging are necessary to enhance system robustness and prevent potential threats.