Project Overview

Project Title: DocuVault: Secure Desktop File Manager (DVFM)

Objective:
DocuVault aims to provide a secure, efficient, and user-friendly desktop application for managing, organizing, and safeguarding digital files. Much like popular cloud-based services but tailored for local use, the system will allow users to store documents, images, and other file types, while offering robust features such as file encryption, categorization, search.

System Architecture & Modules

1. User Authentication & Access Control

- **User Registration & Login:**
  Users create accounts with a username and a strong password. Passwords will be securely hashed (using libraries like bcrypt) and stored in an SQLite database.
- **Session Management:**
  Upon successful login, the application creates a secure session. An auto-lock feature will log out inactive users, thereby safeguarding sensitive data.

2. File Management and Organization

- **Local File Operations:**
  The core functionality will include uploading, downloading, renaming, and deleting files. Python's built-in modules (os, shutil) will facilitate these operations.
- **File Categorization:**
  Users can organize files into custom folders, apply tags, and sort files by type, date, or custom attributes. An auto-sorting mechanism will help classify files based on file type and metadata.
- **Search and Retrieval:**
  An advanced search feature will enable users to locate files by name, type, tags, or date. This will include both basic text search and filter options to narrow down large file collections.

3. File Security and Encryption

- **Data Encryption:**
  To protect sensitive data, files will be encrypted using AES encryption (via PyCryptodome). The encryption key will be securely generated and stored (or derived from user credentials) to prevent unauthorized access.
- **Secure Metadata Storage:**
  All metadata related to files, including file paths, tags, and user activity logs, will be stored in an encrypted SQLite database.

4. User Interface and Experience

- **Desktop GUI:**
  The application will feature a modern, intuitive GUI developed using Tkinter. The interface will include a dashboard displaying recent files, folder navigation, and status notifications.
- **Responsive Design:**
  Even though the application is desktop-based, careful design considerations will ensure that it can adjust to different screen resolutions and usage patterns.

5. Additional Features

- **Audit and Logging:**
  Every action performed by the user (file upload, deletion, etc.) will be logged. This audit trail will help in troubleshooting and provide an extra layer of security by monitoring unauthorized access attempts.
- **User Feedback and Help System:**
  The application will include a help section, FAQs, and a user feedback mechanism to continuously improve the software based on user input.

6. Optional Features (if time permits)

- **AI-based File Classification & Sorting:**

  Automatically detect file types (text, images, PDFs, etc.) and categorize them based on their content. For example, a PDF could be identified as a novel, textbook or datasheet, and it will be sorted accordingly in the file system.

- **Automation:**

  Once a new file is added to a directory, the system should classify it and move it to the correct folder.