1885 SMI UNIVERSITY

Title of Project: Secure Vault

Group Members	Student ID
Zulfiqar	BIT-24S-024
Muhammad Muzzammil	BIT-24S-035
Hammad	BIT-24S-036
Nehal Ali	BIT-24S-041

Subject: Artificial Intelligence.

Program: BS IT (IT-3A).

Submitted to: Prof. Aqsa Umar.

1. Title of Project

SecureVault – A Command-Line Based Password Manager in Python

2. Explain the Idea of Project

SecureVault is a command-line application designed to securely store and manage a user's various online credentials (website, username, password). It simulates a digital safe, where users can store sensitive login data that is accessible only through a master password. All stored data is encrypted using strong cryptographic algorithms to ensure privacy and data integrity.

The app allows the user to:

- Add new credentials
- View existing saved credentials (after unlocking the safe)
- Store all information locally in encrypted form
- The master password is hashed and stored using a cryptographically secure method, so it's never stored in plain text.
- This tool is especially aimed at users who want a simple, offline, secure way to manage their passwords without relying on external software.

3. Features of the Project

✅ **Command-line User Interface:** Easy-to-navigate terminal-based UI for interaction.

✅ **Master Password System:** Set once, and used to unlock the vault each time.

✅ **Secure Password Storage:** Passwords are encrypted and stored in a local file.

✅ **Password Retrieval:** User can view credentials only after authentication.

✅ **Data Stored in CSV Format**: Lightweight format, optionally later upgradeable to encrypted JSON or database.

✅ **Encryption:** AES-based encryption via Python's cryptography library (Fernet).

✅ **User Input Validation:** Basic checks to prevent malformed data entries.

Similar Work:

Applications like Bitwarden (open-source), KeePassXC, and LastPass are popular password managers. SecureVault is a simplified, educational version of those tools built from scratch using Python.

4. Why I Want to Do This Project (Problem Statement)

As our online presence grows, managing multiple passwords becomes challenging. Many users resort to insecure practices like using the same password for every site, or storing passwords in plain text files. This creates a serious security risk.

The goal of this project is to:

- Build a foundational understanding of secure storage.
- Learn encryption and hashing techniques.
- Gain experience with backend logic, file handling, and user authentication.
- Take a step toward cybersecurity-focused development, aligning with my personal interest in security tools and backend engineering (inspired by open-source pioneers like Linus Torvalds).

5. Algorithm & Data Structure Overview

Algorithms Used

Password Hashing Algorithm: PBKDF2_HMAC (via hashlib) for securely hashing the master password with salt.

Symmetric Encryption: AES encryption (via Fernet) to encrypt and decrypt password data before storage or display.

Input/Output Flow Control: Loop-based menu system (while, if-else) to interact with users and process commands.

Data Structures Used

Dictionaries: Used to structure each password entry (site, username, password).

Lists: To store multiple entries in memory before saving.

CSV File Format: Used to store data persistently, row by row.

Byte Strings: Used during encryption/decryption processes.