# **SECURE PASSWORD MANAGER**

## By – Gargi Patel (24je0267)

* The project focuses on creating a secure password manager using AES-256 as the encryption standard.
* It is basically a command-line tool which allows the user to add(store), retrieve and delete credentials without compromising its integrity. It also helps the user to generate random, yet strong passwords of required length.
* The application starts by automatically creating two files (namely, master\_key.enc and credentials.enc) to store the encrypted master key and the credentials of the users respectively.
* The user is prompted to either sign-up or login to their existing account.
* Sign-up: The user is prompted to enter a user name and a master key for their account. The master key is then encrypted using AES-256 with a predefined key. The encrypted master key is then saved to master\_key.enc file, along with the username.
* Login: The user is prompted to enter the creds (username and master key) of their existing account. If it matches the one stored during sign-up, the user is directed to the next menu to perform further operations. Otherwise, they’re directed back to the main menu.
* After logging in, the user can perform the following functions:

1. Add/Store Creds: The user is prompted to enter the name of the website, their username and the password. The password is then encrypted using the same AES-256 encryption with master key as the encryption key. The encrypted password and the rest of the creds get saved to the credentials.enc file.
2. Retrieve Creds: The user is prompted to enter the name of the website, if an entry for that website exists in the credentials.enc file then it is displayed on the screen. Otherwise, the user is directed back to the menu. Before displaying the creds, the password is decrypted using AES-256.
3. Delete Creds: The user is prompted to enter the name of the website, if an entry for that website exists in the credentials.enc file then it is displayed on the screen. The user is asked for confirmation to delete the displayed entry. If prompted YES, the entry is deleted. Otherwise, the user is directed back to the menu.
4. Generate Password: The user is prompted to enter the length of the required password. A strong password of required length is displayed on the screen as well as copied to the clipboard.

* Working of AES-256:

1. It is a block cipher i.e. it converts the plaintext into a 4x4 matrix of 128 bits (or 16 bytes). All the operations are performed on this matrix.
2. It uses 14 rounds of encryption and in each round the following operations are performed in the block - Add Round keys, Shift Rows, Mix Columns and Substitute Bytes. Except in the last round, in which mix columns operation is not performed.
3. AES-256 requires a 256 bits key for encryption and decryption. This key is expanded using a key expansion function so as to create the round keys.
4. In the project, padding has been used to achieve the required length for the plaintext as well as the key provided by the user.