PASWORD MANAGER

INTRODUCTION

In an increasingly digital world, the need for robust password management solutions has become paramount to safeguarding sensitive information. This project aims to address this need by developing a secure password manager application that offers a user-friendly interface, strong encryption mechanisms, and advanced features such as password generation, organization, and synchronization. By adhering to security best practices, cross-platform compatibility, and compliance with privacy regulations, this password manager ensures the protection of user data while providing convenience and peace of mind.

OBJECTIVES

- To develop and implement robust authentication mechanisms, encryption techniques, and secure storage practices to safeguard user passwords against unauthorized access and data breaches.
- To improve usability with an intuitive interface and essential password management features.
- To ensure cross-platform compatibility for seamless access across devices.
- To maintain compliance with data protection regulations and prioritize user privacy.

ALGORITHM USED

This Project defines a Python class named AESCipher that provides methods for encrypting and decrypting passwords using the AES (Advanced Encryption Standard) algorithm in Cipher Block Chaining (CBC) mode.

AES Encryption/Decryption Steps:

- **Encryption:**
  - The AES.new () method initializes a new AES cipher object with the provided key and mode (CBC).
  - The pad () function pads the password string to ensure its length is a multiple of the AES block size.

- The encrypt () method of the cipher object encrypts the padded password.
- The initialization vector (IV) is encoded as a base64 string (iv = b64encode (cipher. iv). decode('utf-8')).
- The ciphertext is encoded as a base64 string (ct = b64encode(ct_bytes). decode('utf-8')).
- The IV and ciphertext strings are concatenated and returned as the encrypted password.

➨ **Decryption:**

- The IV is extracted from the first 24 characters of the base64-encoded input string.
- The ciphertext is extracted from the remaining characters of the input string.
- The AES cipher is initialized with the key and IV.
- The ciphertext is decrypted using the cipher object.
- Padding is removed from the decrypted plaintext using the unpad () function.
- The decrypted plaintext is returned as a string.

LIBRARIES AND FRAMEWORK USED

1. **Tkinter:** Tkinter is the standard GUI (Graphical User Interface) library for Python. It provides tools for creating windows, buttons, labels, and other GUI elements. In this code, it's used to create the graphical interface for the password manager application.
2. **Pycryptodome:** It is a Python library that provides cryptographic functions and protocols, offers a wide range of cryptographic functions, including encryption, decryption, hashing, key generation, digital signatures, and more.

   Version: Pycryptodome = =3.11.0
3. **Base64:** The base64 module provides functions to encode binary data to ASCII characters and decode back. It's used here to encode and decode the initialization vector and ciphertext in AES encryption.
4. **Sqlite3:** It is a built-in Python module that provides an interface to interact with SQLite databases. SQLite is a lightweight, file-based relational database management system that does

not require a separate server process. It allows you to create, read, update, and delete data from databases using SQL queries.

QUICKSTART

1. Install the requirements.

   pip install -r requirements.txt

2. Start the application. python main.py

OUTPUT

1. AUTHENTICATION MODULE:

Fig – 1. Sign Up

Upon filling the required fields (username, master password, confirm master password, favourite food for password recovery) in the signup form and clicking on the signup button, a popup appears confirming that the user account has been successfully created.

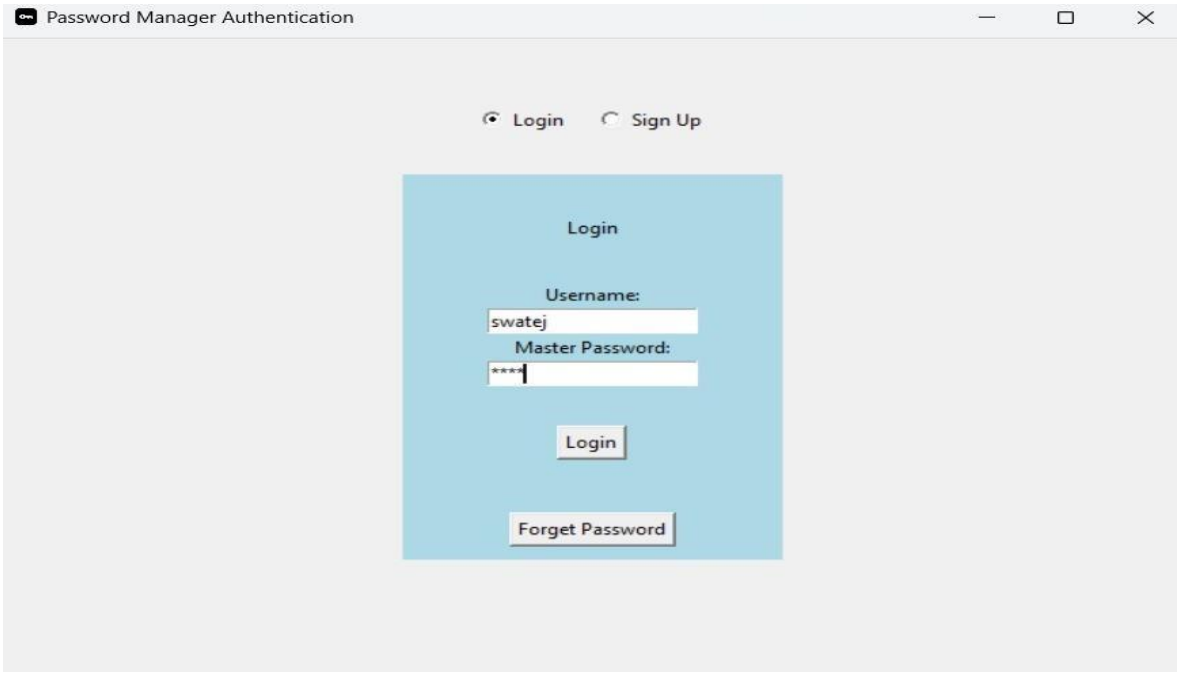


Fig.2 – Login

After entering the credentials, such as username and master password, into the login interface, a popup appears confirming successful login.

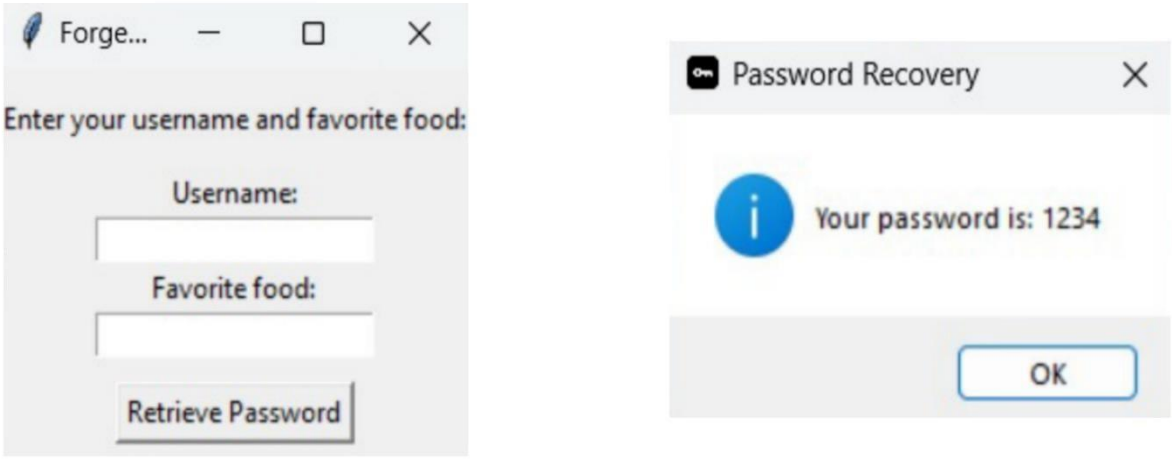


Fig.3 – Password Recovery

If the user fails to recognize the password, they can initiate the password recovery process by entering their username and favourite food into the designated fields.

2. ACCOUNT MANAGEMENT MODULE:

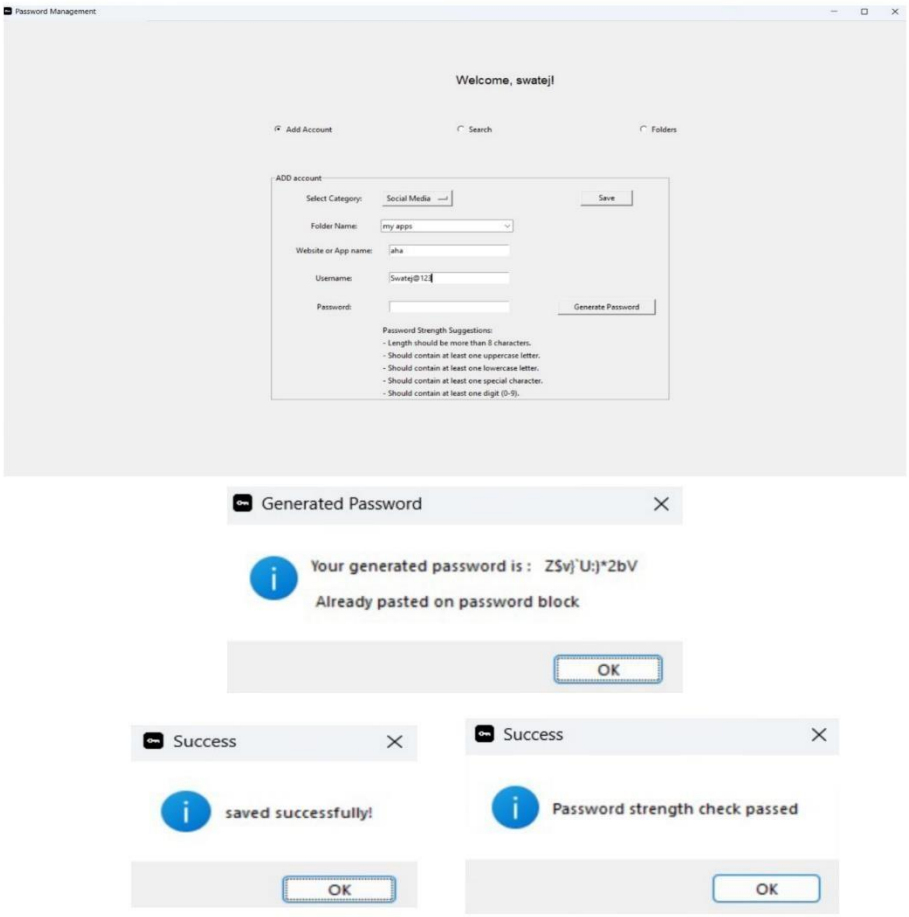


Fig.4 – Add Account

Upon logging in, a new window is displayed with the greeting "Welcome, Username!" Following this, the user selects the "add account" radio button and inputs the folder name, website or app name, and username. Optionally, the user can generate a password by clicking on the "generate" button. Upon completion, clicking the "save" button triggers a popup message confirming the successful saving of the account details.

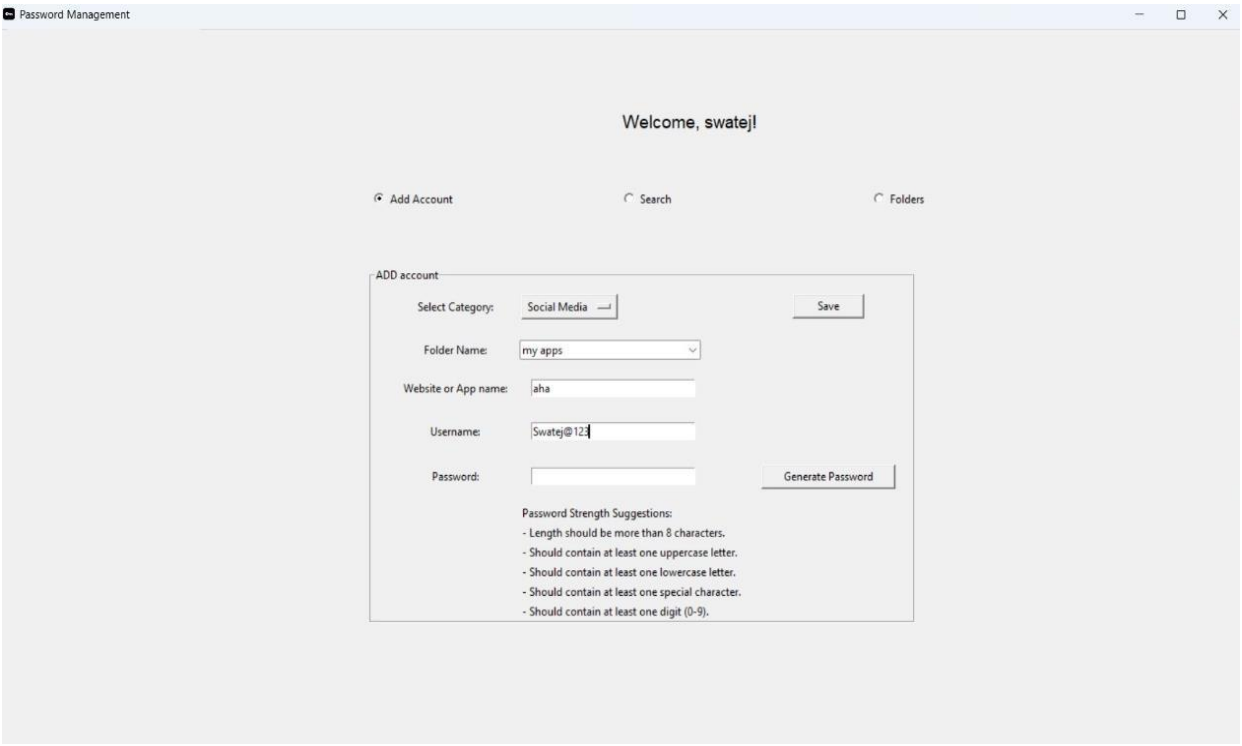


Fig.5 – Selecting Category and Adding Folder Name

The user can add folder names such as "Websites," "Apps," "Mails," "Social media", etc. Depending on the selection of folders, records are stored in the corresponding folders.

3. SEARCH MODULE:

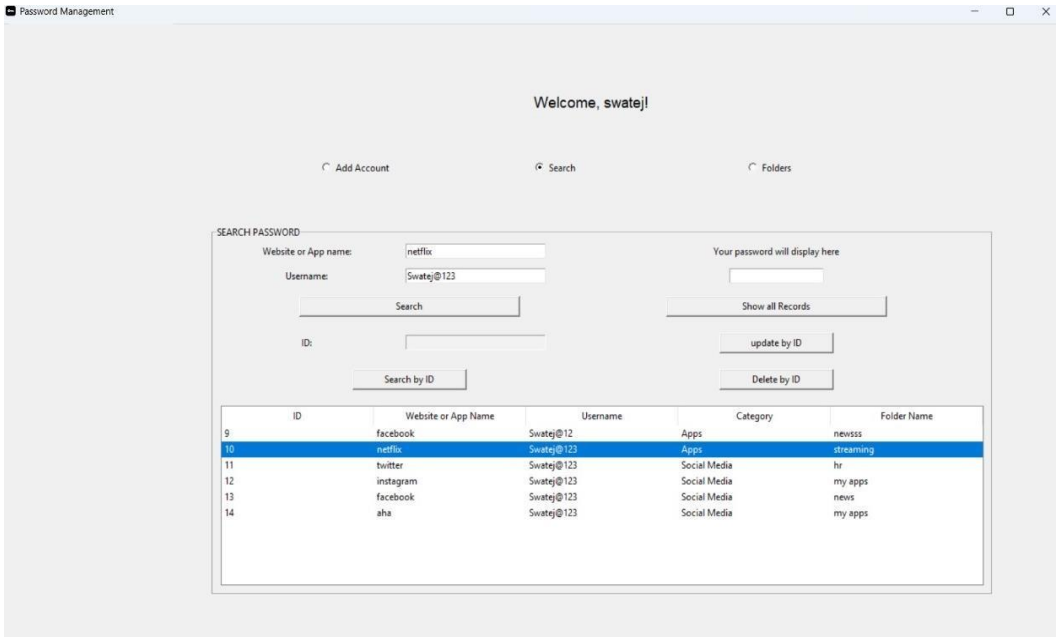


Fig.6 – Search by username

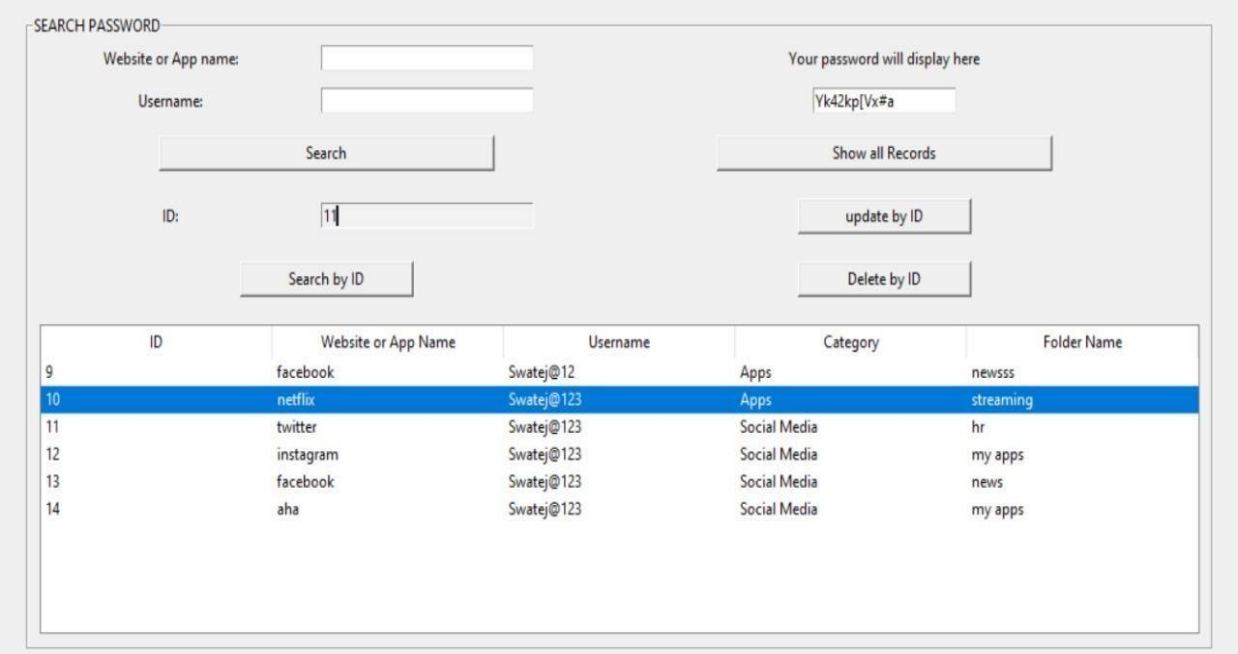


Fig.6 - Search by ID

If the user needs to retrieve a password through the search process, they can do so by entering the username and/or ID. Upon initiating the search, when the user clicks on the website name, the corresponding password is displayed in the password entry box.

4. UPDATE MODULE:

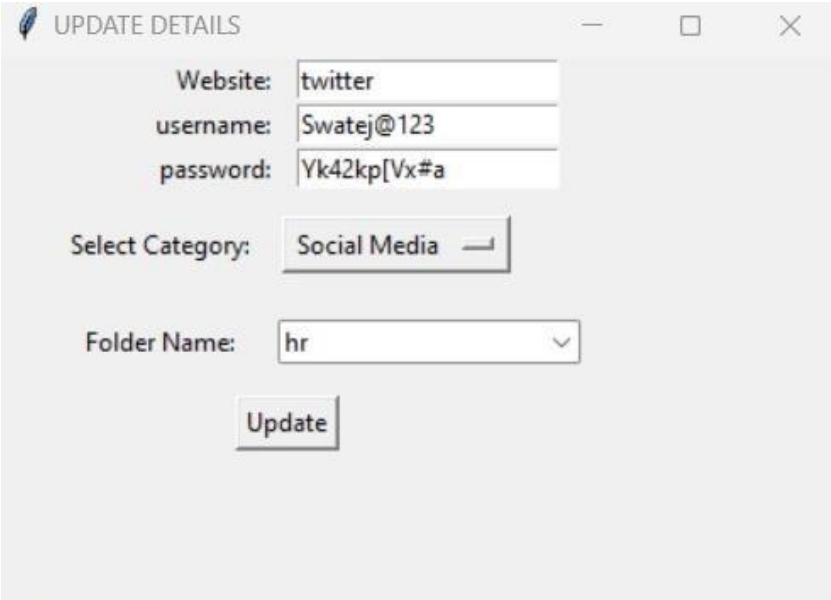


Fig.7 – Update

To update a specific record, the user inputs the ID and clicks on the "Update by ID" button. A new window appears displaying the record details, allowing the user to modify them. After making the desired changes, clicking the "Update" button triggers a message confirming successful updates.

5. DELETE MODULE:

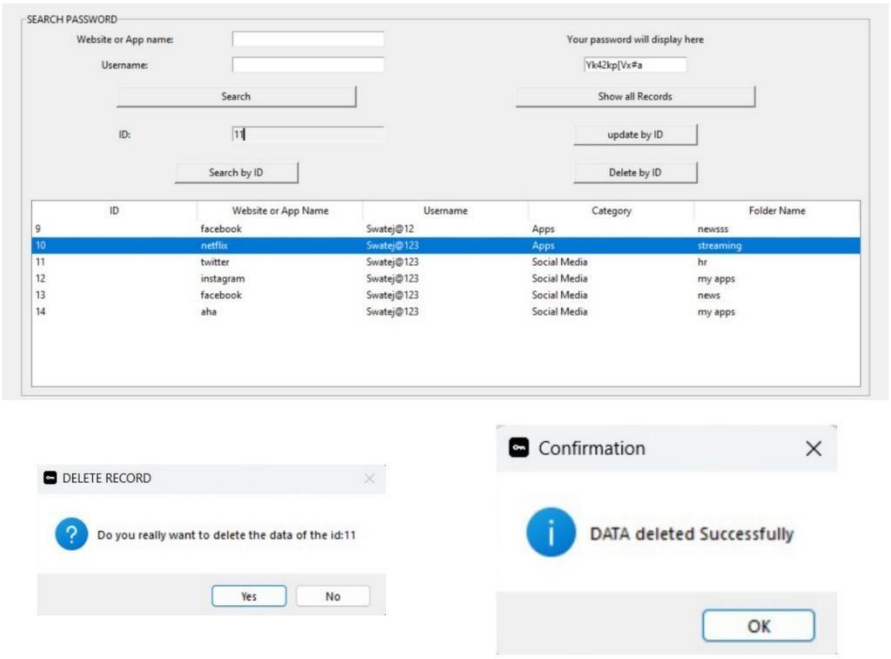


Fig.8 – Delete

To delete a particular record, the user enters the ID and clicks on the "Delete by ID" button. A confirmation popup appears, asking "Do you really want to delete the data of the ID: <id number>?" If confirmed, the record is removed from the list.

6. DISPLAY ALL RECORDS:

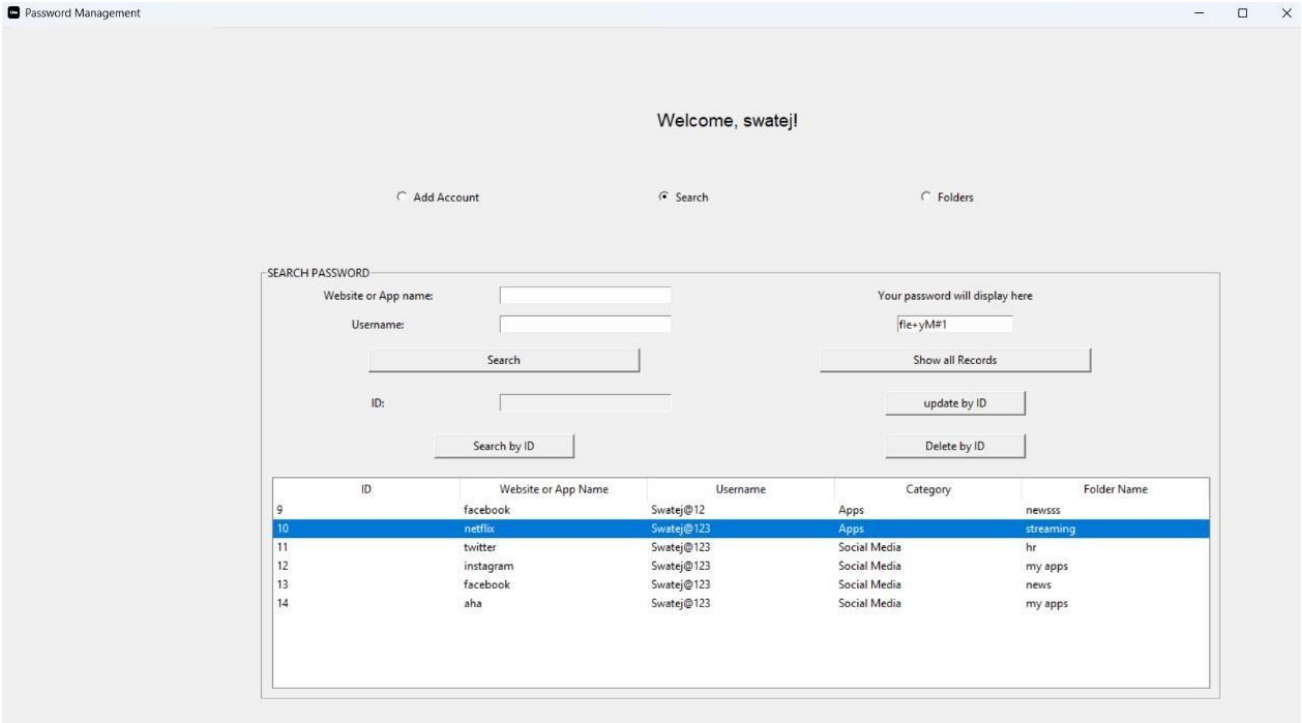
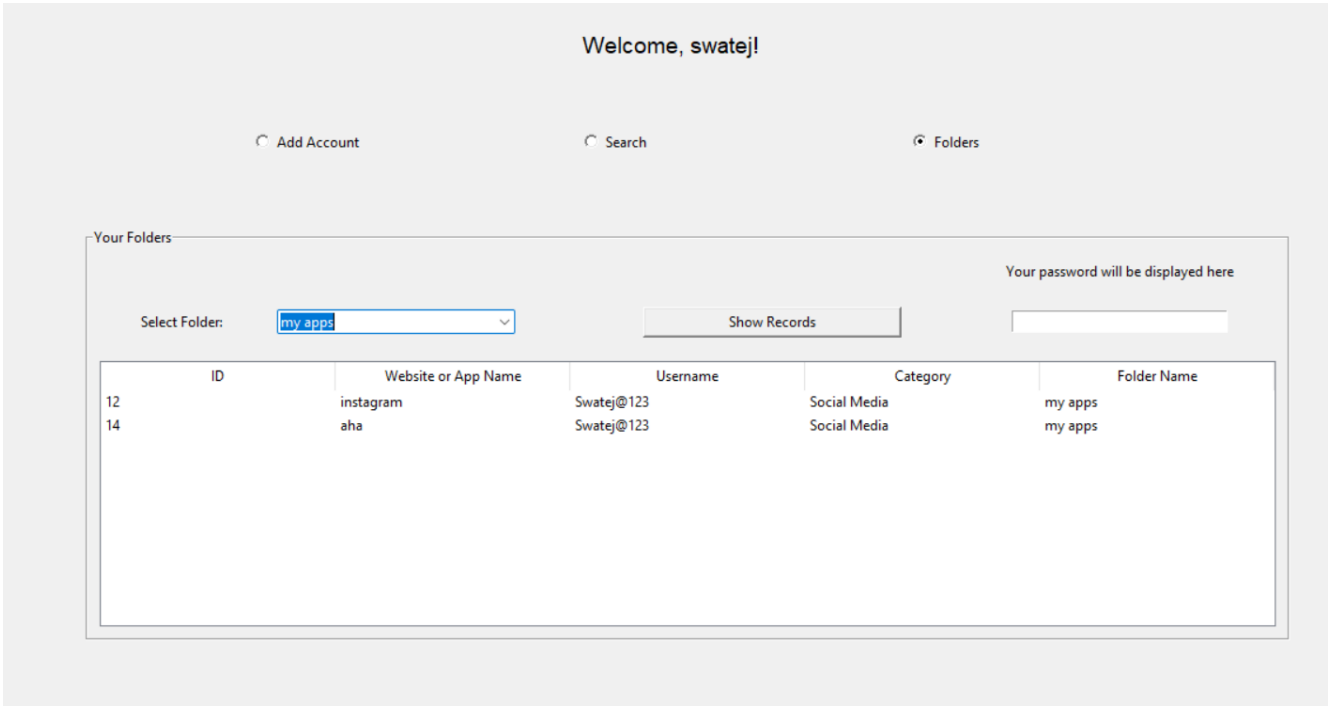


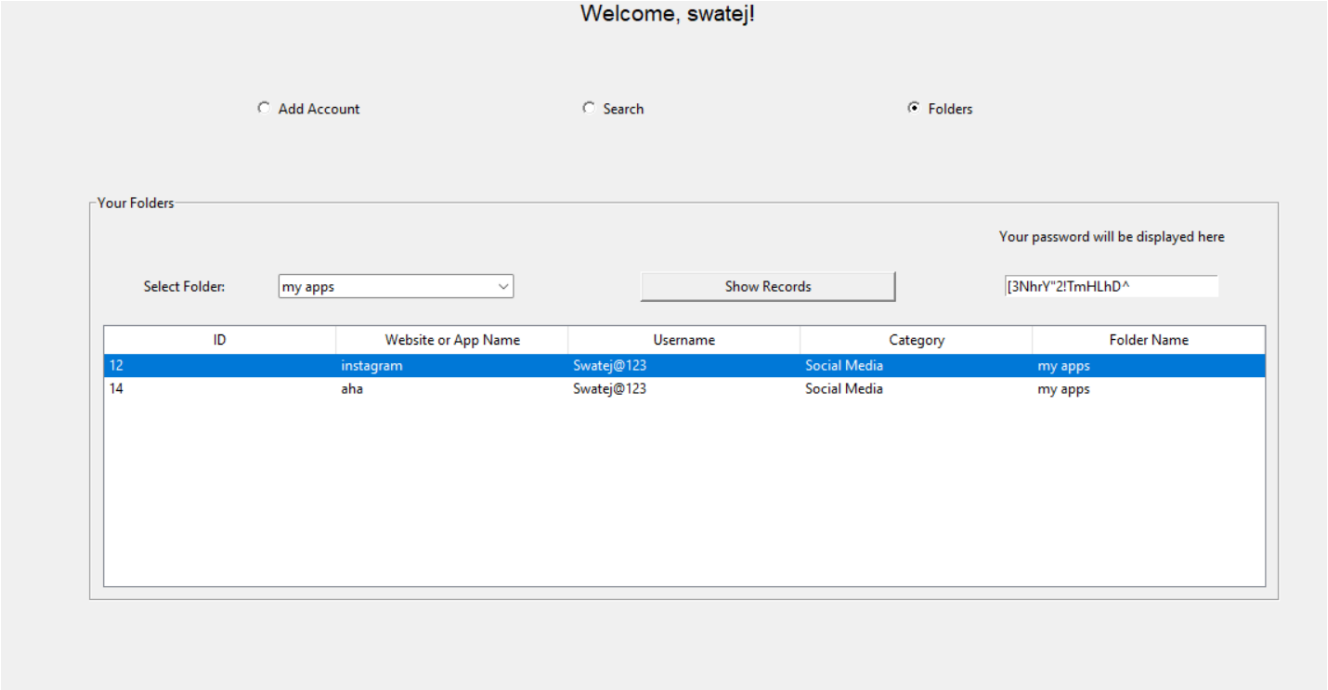
Fig.9 – Show All Records

By clicking on the "Show All Records" button, all records are displayed in a tree view format. If the user clicks on a specific record, the corresponding password is displayed in the password entry box.

7. DISPLAY ALL RECORDS IN FOLDER:



In the above figure we can select any folder name and can view records present on the folder. It helps in easily access of records based on folders. On selecting any the password is automatically displayed in top right corner box.



CONCLUSION

In conclusion, the development of a secure password manager is a complex but essential endeavour to protect users' sensitive information in an increasingly digital world. By implementing features such as strong encryption, authentication mechanisms, password generation, organization, and synchronization, the password manager provides users with a convenient and secure way to manage their credentials across multiple platforms.

Furthermore, adhering to security best practices, ensuring cross-platform compatibility, and regularly updating the application are crucial for maintaining the integrity and reliability of the password manager. Additionally, prioritizing user privacy and data protection, as well as providing comprehensive documentation and support, are essential for ensuring a positive user experience.

Overall, a well-designed and robust password manager can greatly enhance users' security posture and help mitigate the risks associated with password-related vulnerabilities.

*****THE END*****