Extra Credit：簡易密碼管理器實作

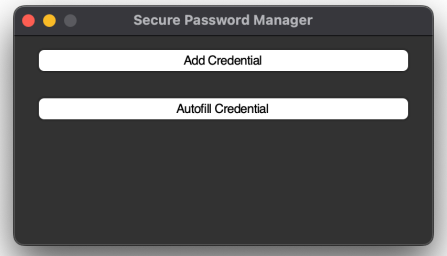
這段 code 實現了一個具有圖形使用者介面（GUI）的簡單密碼管理器原型，使用Python的Tkinter庫來創建。它允許使用者通過圖形介面添加、存儲以及在安全條件下自動填充網站登錄憑證。以下是代碼的詳細解釋和使用步驟：

代碼作用

1. **初始化密碼管理器**：定義了一個SecurePasswordManager類，用於管理使用者的登錄憑證。這些憑證以網站域名（包括協議，如http或https）為鍵，使用者名稱和密碼對作為值，存儲在一個字典中。
2. **添加憑證**：使用者可以通過輸入網站的完整地址（包括協議），使用者名和密碼來添加新的憑證。這些資訊將被存儲在密碼管理器中。
3. **自動填充憑證**：當使用者嘗試自動填充一個網站的憑證時，密碼管理器會檢查提供的域名是否以http://開頭。如果是，出於安全考慮，自動填充將被禁用，並提示使用者。如果是安全的https://連接，且憑證存在，密碼管理器將返回使用者名和密碼。

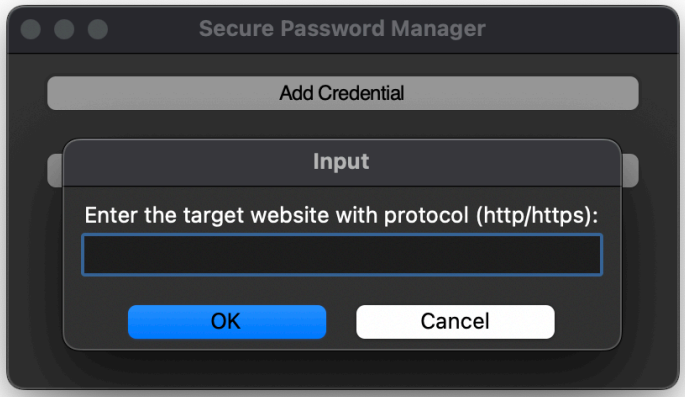
使用步驟

1. **啟動程序**：執行代碼後，會出現一個帶有兩個按鈕的窗口：Add Credential（添加憑證）和Autofill Credential（自動填充憑證）。

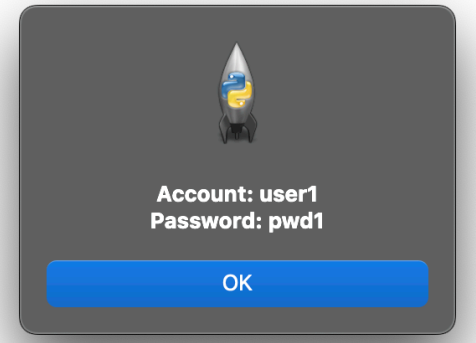


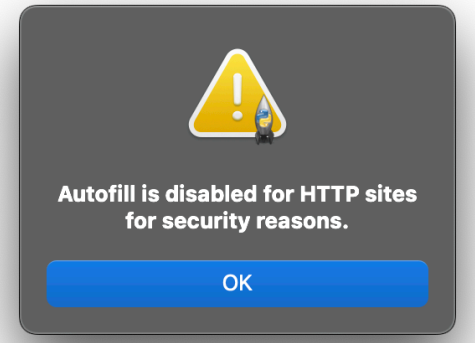
2. **添加憑證**：
    1. 點擊Add Credential按鈕。
    2. 在彈出的對話框中依次輸入目標網站的地址（包括http或https）、帳戶名和密碼。
    3. 提交後，資訊將被儲存，同時彈出一個提示框，告知使用者憑證已成功添加。

3. **嘗試自動填充憑證**：
    1. 點擊Autofill Credential按鈕。
    2. 在彈出的對話框中輸入要自動填充的網站地址（包括http或https）。



    3. 程序會檢查輸入的地址是否符合安全填充的條件：
        1. 如果是http://開頭，將顯示一個警告，告知使用者不能自動填充。
        2. 如果是https://且憑證存在，將顯示帳戶名和密碼。
        3. 如果域名不匹配或使用HTTP，將提示錯誤。





注意事項

1. **安全性**：出於演示目的，密碼在此原型中未進行加密處理。在實際應用中，應確保對敏感資訊進行加密。
2. **HTTP和HTTPS**：自動填充功能只在使用HTTPS的網站上啟用，以保護使用者數據的安全。
3. **GUI介面**：通過Tkinter創建，包括基本的文字輸入和提示框，使使用者操作直觀簡單。

這個原型提供了一個基礎框架，展示了如何使用Python和Tkinter創建一個具有基本功能的密碼管理器GUI應用。在開發完整的應用時，可能需要添加更多高級功能和安全措施，如密碼加密、多因素認證等。