**Algorithm Explanation**

1. **Criteria Definition:**  
   The password\_strength function checks five criteria:
   * **Length:** At least 8 characters (len(password) >= 8)
   * **Lowercase Letter:** Contains at least one lowercase letter ([a-z])
   * **Uppercase Letter:** Contains at least one uppercase letter ([A-Z])
   * **Digit:** Contains at least one number (\d)
   * **Special Character:** Contains at least one special character ([!@#$%^&\*()-\_+=])
2. **Scoring:**  
   Each satisfied criterion adds 1 point to the total score (max score = 5).
3. **Strength Classification:**
   * **Strong:** Meets all 5 criteria
   * **Moderate:** Meets 3–4 criteria
   * **Weak:** Meets fewer than 3 criteria
4. **User Feedback:**  
   A result string summarizes the strength and shows which criteria passed or failed using ✔ and ✖.
5. **GUI Interaction:**  
   The Tkinter GUI allows users to enter a password and receive feedback through a pop-up message.

**Effectiveness of the Algorithm**

**Strengths:**

* **Simple and Clear:** Easy to understand and provides immediate feedback.
* **Comprehensive Basic Checks:** Covers the most common security best practices.
* **User-Friendly:** GUI makes it accessible for non-technical users.

**Limitations:**

* **No Entropy Analysis:** Doesn't evaluate password randomness or predictability.
* **No Dictionary Check:** Can't detect common or compromised passwords.
* **No Length Scaling:** Treats all passwords 8+ characters the same, not rewarding longer ones.

**Overall Effectiveness:**  
This tool effectively encourages strong password habits for general users but may not detect sophisticated vulnerabilities like predictable patterns or dictionary-based weaknesses.

Program file – pss.gui (Documents/Python programming)