Fachhochschule Nordwestschweiz
Hochschule für Wirtschaft

Studierendenprojekt

Programmierprojekt
Passwortgenerator/ -manager

Modul	Grundlagen Programmierung
Autorinnen	Adea Mala Medina Senderovic Leonie Schaub
Dozenten	Hermann Grieder & Dr. Felix Härer
Ort, Datum	Olten, 10.10.2025

Inhalt

1. Projektübersicht

Das Ziel in unserem Projekt ist es einen Passwortgenerator zu erstellen, welcher zufällig Passwörter generiert. Dabei können die Benutzenden die Länge und die Art (Buchstaben, Zahlen, Wörter, Sonderzeichen) des Passwortes selbst festlegen. In einer CSV-Datei werden die Passwörter gespeichert und verwaltet. Unsere Anwendung ist für Benutzende gedacht, welche grossen Wert auf ihre Datensicherheit legen und ihre Passwörter übersichtlich verwalten möchten.

2. Motivation / Begründung

Wir haben uns für diese Anwendung entschieden, da wir selbst Passwortgeneratoren verwenden. In der heutigen Zeit ist es von grosser Bedeutung, sichere Passwörter für die eigenen Accounts zu verwenden. Für uns ist es spannend zu wissen, wie genau ein solcher Passwortgenerator und -manager im Detail aufgebaut ist und wie dieser programmiert wird. Zudem können wir durch dieses Projekt, das im Unterricht gelernte praktisch anwenden.

3. Zielgruppe

Unsere Anwendung ist für Benutzende gedacht, welche grossen Wert auf ihre Datensicherheit legen und ihre Passwörter übersichtlich verwalten möchten. Die Benutzenden können ihre bestehenden Passwörter prüfen, löschen oder neue generieren lassen und diese dann speichern.

4. Hauptfunktionen und Programmablauf

4.1 Funktionsübersicht

Die Anwendung bietet folgende Hauptfunktionen im Hauptmenü:

- Frage nach neuem Passwort
- Passwort prüfen
- Anzeige der Passwortstärke:
  - stark
  - mittel
  - schwach
- Passwort ändern oder löschen
- Einstellbare Passwortlänge zwischen 12 und 15 Zeichen
- Zusammensetzung des Passwortes (Wörter, Gross- und Kleinbuchstaben, Zahlen, Sonderzeichen)
- Passwort anzeigen
- Speichern des generierten Passworts
- Passwort erneut generieren
- Bestätigungsabfrage zur Übernahme der Änderung
- Auswahl des zu löschenden Accounts
- Zweite Sicherheitsabfrage zur Bestätigung der Löschung
- Abfrage nach Löschung, um neues Passwort zu generieren
- Beenden und Verlassen des Programms

4.2 Programmablauf

HAUPTMENÜ (Optionen 1-7)

↓

Benutzer wählt Option

↓

Ausführung der gewählten Funktion

↓

Speichern aller Daten

↓

Rückkehr zum HAUPTMENÜ (Schleife läuft bis zum Beenden)

↓

Programmende

4.3 Wichtige Benutzerinteraktionen

Funktion: sichere und komplexe Passwörter generieren

Der Benutzende wird im Hauptmenü gefragt, ob dieser ein bestehendes Passwort prüfen, löschen oder ein neues erstellen möchte.

Bestehendes Passwort prüfen oder ändern

Wenn der Benutzende das bestehende Passwort auf dessen Sicherheit prüfen möchte, wird das Passwort gemäss den vorgegebenen Kriterien (Länge, Zahlen, Sonderzeichen, Gross- und Kleinbuchstaben) geprüft. Entspricht das Passwort den Anforderungen, wird «stark» ausgegeben und dass das Passwort so belassen werden kann. Weist das Passwort Sicherheitsmängel auf, wird je nach Zusammensetzung des Passwortes «mittel» oder «schwach» ausgegeben. Ob das bestehende Passwort geändert werden soll, teilt der Benutzende mit einer ja/nein-Eingabe mit. Falls der Benutzende das Passwort nicht ändern möchte, wird dieser nochmals gefragt, ob keine Änderung vorgenommen werden soll. Wünscht der Benutzende keine Änderung, wird das Programm verlassen. Bei der Änderung des bestehenden Passwortes wird der Benutzende gefragt, wie das Passwort zusammengesetzt werden sollte. Dabei ist eine Kombination aus Gross- und Kleinbuchstaben, Sonderzeichen, Wörtern und Zahlen möglich. Das erstellte Passwort wird dann in der CSV-Datei abgespeichert.

Neues Passwort erstellen

Der Benutzende kann im Hauptmenü auswählen, dass dieser ein neues Passwort wünscht. Dabei wird das Passwort nach den Präferenzen des Benutzenden zusammengestellt. Nach der Erstellung des Passwortes muss der Benutzende eingeben, für welche Applikation dieses Passwort verwendet werden soll. Das aktualisierte Passwort wird dann in die CSV-Datei importiert.

Passwort löschen

Benötigt der Benutzende ein Passwort für eine Applikation nicht mehr, kann dieser das Programm auffordern, dieses zu löschen. Das Passwort und die Applikation werden gelöscht und die Aktualisierung wird in die CSV-Datei überführt. Anschliessend wird das Programm verlassen.

Funktion: Speicherung und Aktualisierung in einer CSV-Datei

Die geänderten, gelöschten oder neu erstellten Passwörter werden in eine CSV-Datei importiert. In dieser Datei sind die Passwörter übersichtlich dargestellt. Der Benutzende sieht für welche Applikation welches Passwort verwendet wird.

5. Erforderliche Kriterien

5.1 Interaktive Anwendung

Der Benutzende wird nach dem Programmstart gefragt, ob dieser ein neues Passwort erstellen, löschen oder ein bestehendes prüfen möchte. Der Benutzende gibt ein, welche Funktion das Programm durchführen soll. Möchte der Benutzende ein neues Passwort erstellen oder will dieser ein bestehendes Ändern, muss dieser die Länge des Passwortes und dessen Zusammensetzung eingeben. Danach wird das Passwort zufällig generiert. Der Benutzende bestimmt mit einer ja/nein-Eingabe, ob das vorgeschlagene Passwort übernommen werden oder nochmals ein neues generiert werden soll. Anschliessend wird das Passwort in eine Datei importiert. Möchte der Benutzende ein Passwort löschen, kann dieser die Applikation auswählen. Das Passwort und die Applikation werden danach gelöscht und die Änderung wird in die CSV-Datei überführt.

5.2 Datenvalidierung

Das Programm prüft die Länge und die Zeichenvielfalt eines bestehenden Passwortes. Dabei wird geprüft, ob das Passwort die festgelegte Zeichenlänge von 12 bis 15 Zeichen erfüllt. Zudem darf das Passwort keine Wiederholungen oder Aneinanderreihungen von Zeichen enthalten. Ist die Zeichenlänge zu kurz/lang oder gibt es Aneinanderreihungen und Wiederholungen, wird im Terminal «schwach» ausgegeben. Ist eines der zu prüfenden Kriterien nicht erfüllt, wird im Terminal «mittel» ausgegeben. Der Benutzende kann das Passwort ändern.

Zudem wird geprüft, ob ein Passwort für mehrere Applikationen verwendet wird. Ist dies der Fall, muss der Benutzende die Passwörter ändern.

Bei unvollständigen oder inkorrekten Eingaben erhält der Benutzende eine Fehlermeldung, die diesen darauf hinweisen, dass nicht alle Kriterien erfüllt wurden. Zur Überprüfung der Eingaben werden if-elif-else-Bedingungen genutzt. Dadurch kann das Programm erkennen, ob die Anforderungen erfüllt sind, bevor das Passwort erstellt und gespeichert wird.

5.3 Dateiverarbeitung

Beim Programmstart wird die CSV-Datei entschlüsselt und in Python eingelesen. Die in der CSV-Datei enthaltenen Daten werden danach in Dictionaries und Listen umgewandelt. Bei einer CSV-Datei entspricht jede Zeile einem Datensatz. Nach dem Einlesen wird dann die Hauptfunktion des Programmes durchgeführt, welche aus dem Generieren, Prüfen oder Löschen von Passwörtern bestehen. Durch das Modul «secrets» werden zufällige Passwörter generiert, welche schwer zu erraten sind. Wichtig ist, dass jeder Befehl fehlerfrei ausgeführt wird, damit keine Datensätze verloren gehen oder beschädigt werden. Nachdem die Hauptfunktonen durchgeführt wurden, werden die aktualisierten Daten in das ursprüngliche CSV-Format überführt. Zuletzt hat der Benutzende eine CSV-Datei, in welcher die Passwörter sicher abgespeichert sind.

6. Technische Anforderungen

- **Programmiersprache**: Python
- **Datentypen:** Strings, Floats, Integers, Listen, Bools
- **Kontrollstrukturen:** if/elif/else, while-Schleifen, for-Schleifen, while-else-Schleife
- **Funktionen:** Separate Funktionen für jede Hauptfunktion (print, break, continue, input)
- **Dateiverarbeitung:** open(), read(), write(), CSV-Handling, import ()
- **Exception-Handling:** try-except für Eingabevalidierung
- **String-Operationen:** Formatierung, Parsing von Daten, E-Mail-Validierung, Length

7. Erwartete Herausforderungen

Das Programmieren des Passwortgenerators und -manager könnte etwas herausfordernd werden, da unsere Gruppe sich erst seit dem Beginn des Studiums mit dem Programmieren befasst. Jedoch können wir mit diesem Projekt, das im Unterricht gelernte festigen, indem wir die Codes programmieren und dadurch ein besseres Verständnis für das Programmieren erhalten. Damit das Programm alle unsere Voraussetzungen berücksichtigt, braucht es verschiedene Funktionen, welche fehlerfrei ausgeführt werden müssen. Die grösste Herausforderung wird es sein, Fehler, die sich beim Programmieren eingeschlichen haben, zu beheben. Zudem könnte eine weitere Herausforderung das Programmieren der Abfrage der bestehenden Passwörter sein, denn wenn ein bestehendes Passwort geändert wird, sollte dieses dann auch in der CSV-Datei angepasst werden.