LASTENHEFT

Softwareprojekt – Passwort-Manager

Joel, Matthis, Nils, Nico

Inhalt

1. Projektbezeichnung

Entwicklung eines lokalen Passwortmanagers in Python mit sicherer Speicherung und Benutzerverwaltung.

2. Ausgangssituation / Ist-Zustand

Viele Nutzer speichern ihre Passwörter unsicher, z. B. in unverschlüsselten Notizen oder auf Papier. Es fehlt eine einfach zu bedienende, lokal ausführbare Softwarelösung, die eine sichere Passwortverwaltung ermöglicht. Besonders für technisch weniger versierte Anwender ist der Bedarf nach einer vertrauenswürdigen Anwendung hoch, die ohne Cloud-Anbindung funktioniert.

3. Zielsetzung

Ziel ist die Entwicklung eines benutzerfreundlichen Passwortmanagers, der eine sichere Speicherung von Zugangsdaten ermöglicht und durch Zugriffsschutz vor unbefugtem Zugriff schützt. Die Anwendung soll lokal auf einem Rechner lauffähig sein, mit einer SQL-Datenbank im Hintergrund arbeiten und datenschutzrechtliche Anforderungen berücksichtigen.

3. Aufteilung (Rollen)

Die Teammitglieder übernehmen jeweils spezifische Rollen zur Projektorganisation und setzen eigenverantwortlich Teilaufgaben um. Die folgende Tabelle zeigt die Aufgabenverteilung:

Rolle	Verantwortlicher	Aufgabenbereich
Zeitwächter	Joel	Überwachung der Zeitplanung
Moderator	Matthis	Koordination der Zusammenarbeit
Protokollant	Nico	Dokumentation der Sitzungen
Statusmanager	Nils	Fortschrittüberwachung

Technische Aufgabenverteilung:

Aufgabe	Verantwortlicher
GUI & Navigation	Nils
Verschlüsselung & Speicherung	Nico
Passwortgenerierung	Joel
Passwort-Sicherheitsscheck	Matthis

4. Muss-Kriterien

1. Passwortdatenbank mit Benutzer-Login
   1.1. Beim Start der Anwendung erscheint ein Login-Fenster. Nur nach erfolgreicher Authentifizierung wird der Zugriff auf gespeicherte Passwörter ermöglicht.
2. GUI für alle Benutzereingaben und zur Passwortverwaltung
   2.1. Speichern, Bearbeiten und Löschen von Passworteinträgen
3. Verbindung zu MySQL-Datenbank
   3.1. Speicherung der Passwörter erfolgt verschlüsselt, z. B. mit AES (Advanced Encryption Standard) oder ähnlichem Verfahren.
4. Benutzerverwaltung
   4.1. Es soll nur einen Benutzer geben, sein Passwort soll allerdings anpassbar sein.

5. Wunsch-Kriterien

- Passwortgenerator
- Passwort-Sicherheitsbewertung
- Export-Funktion (verschlüsselte Datei)

6. Nicht-Ziele

- Keine Cloud-Anbindung
- Keine mobile App

7. Qualitätsanforderungen

- Benutzerfreundliche GUI
- Datensicherheit und Datenschutz
- Stabilität und einfache Installation
- Wartbarkeit des Codes

8. Randbedingungen

- Programmiersprache: Python 3.x
- Datenbank: MySQL / SQLite
- Open-Source Bibliotheken erlaubt (z. B. cryptography, pymysql)
- Projektzeitraum: bis 11. Juni, Zwischenstand 22. Mai

9. Risiken und Annahmen

- Risiken: Verzögerungen bei der Implementierung der Verschlüsselung, Probleme mit GUI-Bibliotheken
- Annahmen: Alle Teammitglieder haben Zugriff auf Entwicklungsumgebung mit Python, MySQL und IDE