



Fachhochschule Nordwestschweiz
Hochschule für Wirtschaft

Studierendenprojekt

GradeCalc

Grundlagen Programmierung

ZIEL DES PROJEKTES

HAUPTZIELE

Benutzerverwaltung mit **Login und Registrierung**

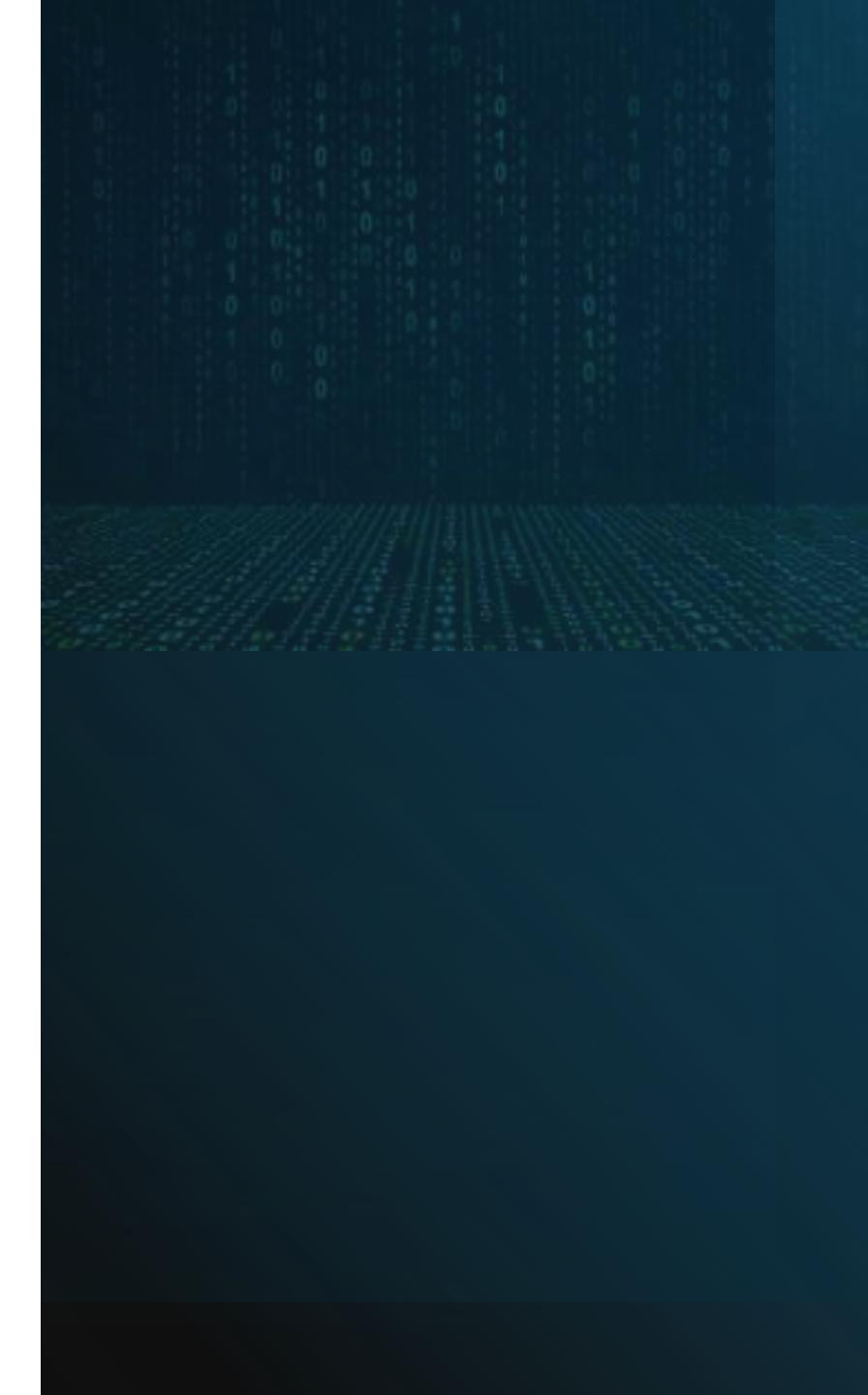
Noten **einfach erfassen und speichern**

Durchschnitte berechnen, mit und ohne Gewichtung

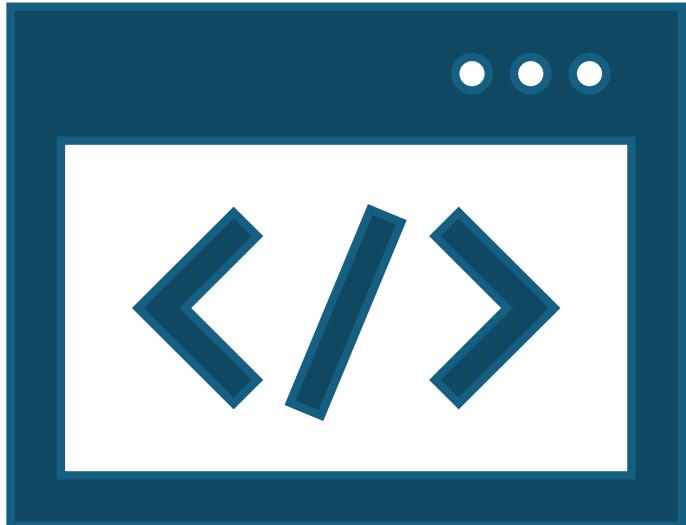
Noten aus Punkten berechnen

Notenskalen umrechnen (Schweiz, Deutschland, USA)

Saubere Fehlerbehandlung bei falschen Eingaben



AUFBAU UND STRUKTUR



Das Programm ist **modular aufgebaut**, das heisst:

- Jedes Modul hat eine **klare Aufgabe**
- Hauptmenü, Logik und Datenhaltung sind getrennt
- Wiederverwendbare Hilfsfunktionen (z.B. Farben, Stilmittel, säubern des Terminals)

Wichtige Module sind zum Beispiel:

- main_menu – Startmenü (Gastmodus)
- user_menu – Menü für eingeloggte Benutzer
- durchschnitt_berechnen
- notenrechnung
- notenskala
- Module für **Noten – und Modulverwaltung**

DATENHALTUNG

Die Benutzerdaten werden in einer JSON – Datei gespeichert

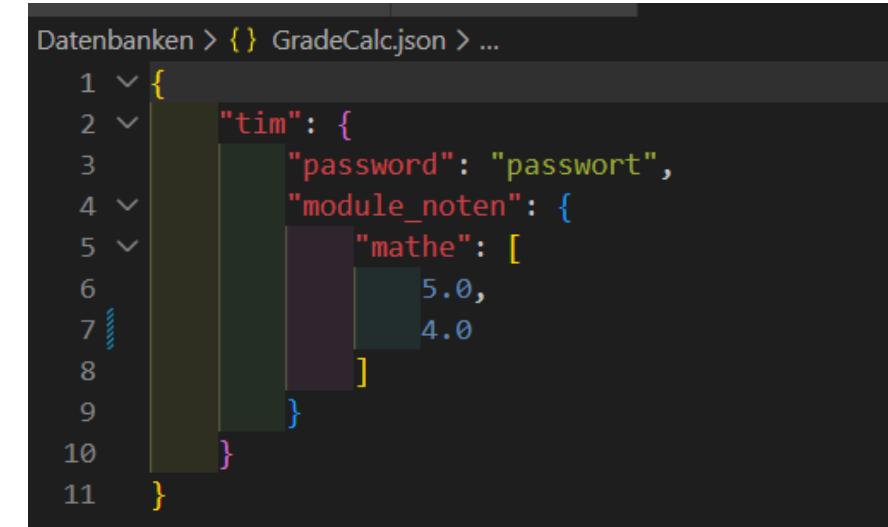
Struktur:

Jeder Benutzer hat:

- Ein Passwort
- Ein Benutzername
- Mehrere Module
- Jedes Modul enthält eine Liste von Noten

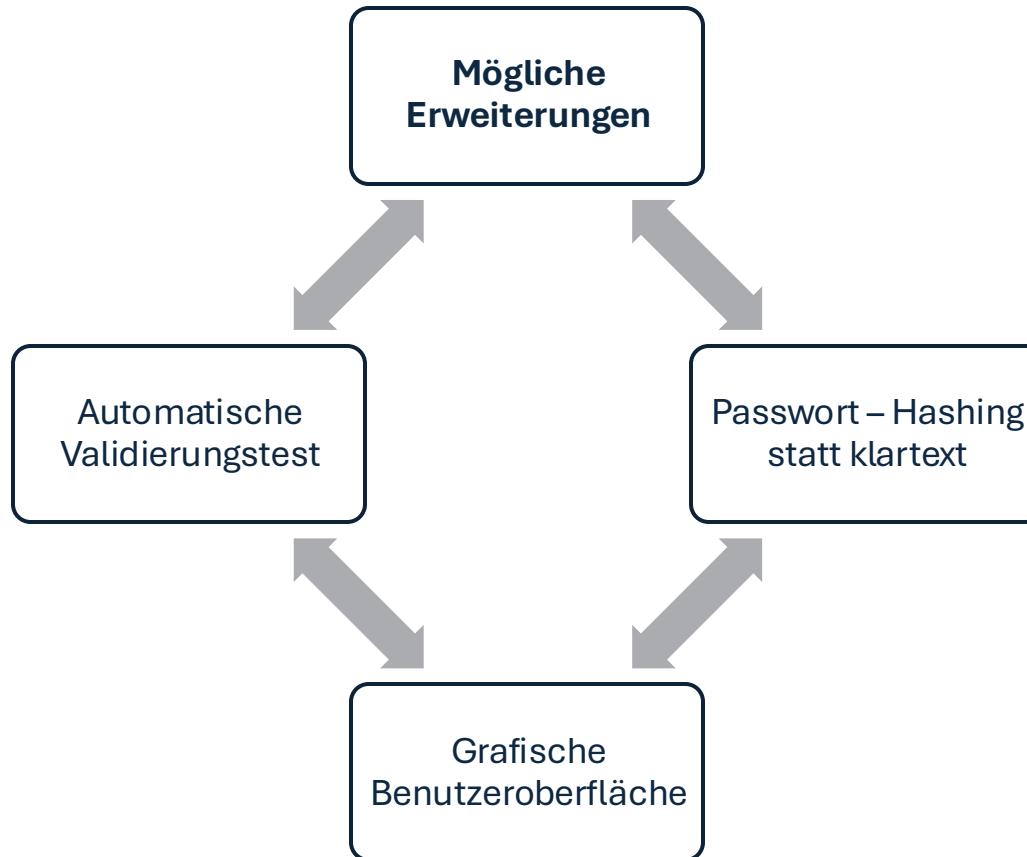
Warum JSON?

- Leicht lesbar
- Keine externe Datenbank nötig
- Perfekt für kleine bis mittlere Datenmenge
- Direkte mit Python kompatibel
- Ohne Datenbank overhead



```
Datenbanken > {} GradeCalc.json > ...
1 < {  
2   < "tim": {  
3     "password": "passwort",  
4     "module_noten": {  
5       "mathe": [  
6         5.0,  
7         4.0  
8       ]  
9     }  
10   }  
11 }
```

VERBESSERUNGSMÖGLICHKEITEN



PROJEKTMANAGEMENT

Python Konsolenprogramm GradeCalc – Team Projekt

Wer macht was?

Damit wir parallel arbeiten konnten, haben wir die Aufgaben klar untereinander aufgeteilt:

Tim :

- Anmelden.py (Login/Registrierung)
- Deine_Module.py (Anzeige der Module und Noten)
- User_menu (Menü für eingeloggte Benutzer)
- Module_hinzufügen_entfernen (Module verwalten)

Irfan

- Notenberechnung (Punkte → Note)
- Noten_hinzufügen_entfernen (Noten verwalten)
- durchschnitt_berechnen (Durchschnitt normal und gewichtet)
- ausserdem das **README** (Dokumentation)

Josselyn:

- ANSI (Farben)
- notenskala (Noten-Umrechner)
- Main_menu / Main_programm (Startmenü und Navigation)
- sowie die Struktur und Pflege der **GradeCalc JSON-Datei**

Mit welchen Mitteln?

Wir haben folgende Mittel genutzt:

Python als Programmiersprache

Visual Studio Code zum Schreiben und Testen des Codes

Git und **GitHub** zur Versionsverwaltung und Zusammenarbeit

JSON-Datei zur Datenspeicherung, da sie leicht lesbar und für ein Schulprojekt gut ist

Terminal/Konsole zum Ausführen und Testen des Programms

ChatGPT Hilfsmittel

Was kann schiefgehen?

Im Projekt haben wir typische Risiken berücksichtigt:

- **Falsche Eingaben** (z.B. Buchstaben statt Zahlen, Noten ausserhalb 1–6)
→ Lösung: try/except, Range-Checks, Wiederholungsschleifen
- **Programmabstürze wegen Datei-Problemen** (fehlende oder kaputte JSON-Datei)
→ Lösung: Datei prüfen/erstellen, sauberes Laden/Speichern
- **Login-Probleme durch Gross-/Kleinschreibung**
→ Lösung: Eingaben von **USERNAME** immer mit Kleinschreibung (z.B. .lower ())
- **Umrechnung der Notenskalen** ist nicht überall gleich standardisiert
→ Lösung: Umrechnung als lineare Approximation deklarieren, Grenzen definieren
- **Merge-Konflikte und doppelte Änderungen** bei Teamarbeit
→ Lösung: klare Aufgabenteilung, regelmässig push/pull, saubere Commits

Qualität sichern

Zur Qualitätssicherung haben wir folgende Punkte eingesetzt:

- **Tests mit Risikostelle**, z.B. 0 Punkte, maximale Punkte, leere Notenlisten
- **Fehlerbehandlung**, damit das Programm bei falschen Eingaben nicht abstürzt
- **Kommentare im Code** für bessere Verständlichkeit
- **Docstrings** für bessere Verständlichkeit von Funktionen und Klasse
- ein **README**, das Installation und Nutzung erklärt



Fachhochschule Nordwestschweiz
Hochschule für Wirtschaft

Studierendenprojekt

TIM
IRFAN
JOSSELYN



BESTEN DANK