

EIBE Projekt-Bewertung

1. Präsentation, Design, Konzept: _ / 40%

1.1 Benutzerfreundlichkeit und Gestaltung: _ / 15%

Intuitive Bedienung: _ / 5

Angemessenes Feedback an die User: _ / 5

Ansprechende Darstellung: _ / 5

Beispiel: Das ist eine schöne Konsolenanwendungen! Die Menüführung ist solide, wobei unsinnige Eingaben erkannt werden und als falsch gelten. Sie könnten die Darstellung noch weiter aufwerten mit Emojis und farbigem Text (ANSI-Codes).w

1.2 Originalität der Projektidee: _ / 10%

Niveau	Beschreibung
5 - Standard	Umsetzung eines der vorgegebenen Konzepte. Funktionalität steht im Vordergrund, wenig eigene Elemente.
7 - Erweitert	Eigenständige Weiterentwicklung mit deutlichen individuellen Elementen. Kreative Anpassungen und eigene Ideen sind erkennbar.
10 - Herausragend	Eigenständiges Konzept oder innovative Weiterentwicklung. Zeigt besondere Kreativität, ungewöhnliche Lösungsansätze oder neuartige Anwendungsfälle.

1.3 README-Datei: _ / 15%

- Erklärt, was das Programm ist: _/5

- Beschreibt, wie man das Programm ausführt: _/5

- Deklariert den Einsatz von KI präzise: _/5

2. SQL-Abfragen: _ / 30%

“SQL-Abfragen: Mindestens 5 verschiedene SQL-Abfragen unterschiedlicher Komplexität, 2 davon mit Aggregatfunktionen. / Korrektheit, Komplexität, Vielfalt”

Niveau	Beschreibung
18 - Grundlegend	Variation von SELECT-Anweisungen mit FROM und WHERE. Grundlegende Filterung und Sortierung.
25 - Fortgeschritten	Verwendung von Aggregatsfunktionen (COUNT, SUM, AVG), GROUP BY. Einfügen von Pythondaten in SQL-Abfragen.
30 - Komplex	Nutzt weiterführende Möglichkeiten von SQL (z.B. join, subqueries).

Beispiel: Sie nutzen insgesamt zehn SQL-Abfragen, drei davon mit Aggregatfunktionen. Sie nutzen zudem komplexere Funktionen wie JOIN für Ihre eigens kreierte country-Tabelle, RANDOM(), und setzen dynamisch Daten aus Python in Ihre Queries ein.

3. Python-Code Qualität: _ / 30%

“Korrektheit, Organisation des Codes, Niveau und Fehlerbehandlung/Robustheit”

Niveau	Beschreibung
10 - Grundlegend	Verwendung von Variablen, einfache Operatoren, print()-Anweisungen, einfache if-else Strukturen, grundlegende Eingabe mit input()
18 - Erweitert	Verwendung von Schleifen (for, while), Listen, einfache Funktionen, string-Formatierung
25 - Fortgeschritten	Eigene Funktionen mit Parametern und Rückgabewerten, Dateiverarbeitung, Module importieren
30 - Komplex	Substanzielle, erweiterte Programmierkonzepte genutzt, z.B. Fehlerbehandlung, try-except, Klassen

Beispiel: Sie nutzen die Möglichkeiten des Unterrichtsumfangs kompetent und gehen sogar darüber hinaus. Sie packen Funktionen in eine Tupel-Liste, nutzen random sowie try-except sinnvoll. Ein rundum solides Projekt mit sehr gutem Code – bravo!

Gesamtbewertung: _ / 100%