

# MC-Test App - Einseiten-Ueberblick

## Was es ist

MC-Test ist eine Open-Source Streamlit-Webapp fuer formatives Multiple-Choice-Ueben und die Qualitaetssicherung von MCQ-Items. Sie liefert schnelles Feedback mit Erklaerungen und Glossaren sowie Auswertungen fuer Lernende und Lehrende.

## Fuer wen

Primaer fuer Studierende/Lernende in Informatik- und verwandten Kursen, die MC-Fragen ueben. Zusaetzlich fuer Lehrende und Item-Autorinnen, die Fragensets pflegen, validieren und auswerten.

## Was es kann

- Pseudonymer Einstieg ohne Konto (Namensliste).
- Uebungsmodi mit Navigation, Markieren und Ueberspringen.
- Feedback mit Erklaerungen, optionalen Erweiterungen und Mini-Glossaren.
- Scoring mit plus-only oder plus/minus; optionale Zeitlimits.
- Pacing-Cooldowns inkl. "Panic Mode".
- Analytics/Dashboards nach Topic und Cognitive Level.
- Exporte: PDF, CSV, DB-Dump, Anki (.apkg).

## Wie es funktioniert (Architektur)

- UI: Streamlit-Entry in `app.py`, Views in `main_view.py`, UI-Module in `components.py`, Texte in `i18n/`.
- Daten: Fragensets als JSON in `data/questions_*.json`, geladen via `config.py` (`load_questions`).
- Logik: `logic.py` fuer Scoring/Navigation, `pacing_helper.py` fuer Cooldowns, `auth.py` fuer Session/Pseudonym.
- Persistenz: `database.py` initialisiert SQLite `db/mc_test_data.db` fuer Sessions/Statistiken.
- Admin & Exporte: `admin_panel.py`, `pdf_export.py`, Ordner `exporters/`.
- Datenfluss (aus Repo-Struktur abgeleitet): Fragenset -> UI -> Antworten -> SQLite -> Analytics/Exporte.

## So startest du

1. `pip install -r requirements.txt`
2. `streamlit run app.py`
3. Browser oeffnen: `http://localhost:8501`

Voraussetzung laut Repo: Python 3.10 bis 3.12.