

Quiz Learning

von:

Yazid Heimur

Malik Ceesay





Gliederung

1. Problemstellung, Anwendungsübersicht & Zielgruppe
2. Quiz-System und Funktionen
3. Live Demo
4. Design
5. Technische Architektur
6. Zentrale Entwicklungsentscheidungen
7. Projektstand und Ausblick



1. Problemstellung

Herausforderungen im traditionellen Lernen:

Motivationsprobleme:

- Monotone Lernmethoden ohne interaktive Elemente
- Schwierigkeit, sich zum Lernen zu motivieren

Zeitmanagement:

- Schwierigkeit, kurze Lerneinheiten sinnvoll zu nutzen
- Keine flexiblen Lernmöglichkeiten zwischendurch

Lernfortschrittskontrolle:

- Keine systematische Erfassung des Lernstands
- Fehlende Übersicht über längerfristige Entwicklung

Feedback-Mangel:

- Verzögertes oder fehlendes Feedback zu Lerninhalten
- Keine unmittelbare Bestätigung des Verständnisses



1. Anwendungsübersicht

Unsere Anwendung bietet:

Intelligentes Feedback-System:

- Sofortiges Feedback nach Abgabe der Antworten
- Fortschrittsvisualisierung in Echtzeit

Motivationsfördernde Elemente:

- Gamification durch Punktesystem
- Soziale Interaktion durch Freundschaftssystem

Flexibles Lernmanagement:

- Kurze, effiziente Quizeinheiten (5 Fragen)
- Zeitsparendes Multiple-Choice Format

Umfassende Fortschrittskontrolle:

- Identifikation von Schwachstellen
- Vergleichsmöglichkeiten mit anderen Nutzern



1. Zielgruppe

Zielgruppe & Nutzen:

Schüler und Studierende
aller Fachrichtungen

Berufstätige und Fachkräfte
in Weiterbildung

Selbstlernende und
Wissbegierige

Unternehmen und
Schulungseinrichtungen

2. Quiz-System & Funktionen



Quiz-Struktur :



- Jedes Quiz besteht aus 5 Fragen



- Bewusst kurz gehalten für häufigere Lernsessions



- Zwei Spielmodi: Zufallsquiz oder Kategorieauswahl



- Multiple-Choice Format mit 4 Antwortmöglichkeiten



2. Quiz-System & Funktionen

Timer-System:

Der 15-Sekunden Timer pro Frage:

- Fördert schnelles Denken und instinktive Antworten
- Verhindert langes Nachdenken oder externes Nachschlagen
- Steigert Spannung und Spielcharakter
- Keine Antwortabgabe nach Zeitablauf möglich

2. Quiz-System & Funktionen

Auswertungsmechanik:

- Detaillierte Auswertung nach Quizende
- Erreichte Punktzahl
- Anzeige der richtigen und falschen Antworten

Quiz Ergebnis

4/5 Punkte

80% richtig beantwortet

Welche Datenbank wird oft mit Flask für kleine Projekte verwendet?

Deine Antwort: SQLite

Wie können Fehlerseiten (z. B. 404) in Flask behandelt werden?

Deine Antwort: Mit der Methode `@app.errorhandler()`

Was bewirkt `app.run()` in Flask?

Deine Antwort: Initialisiert die Anwendung
Richtige Antwort: Startet den Webserver

Was ist der Standardport für Flask-Anwendungen?

Deine Antwort: 5000

Welche Template-Engine verwendet Flask?

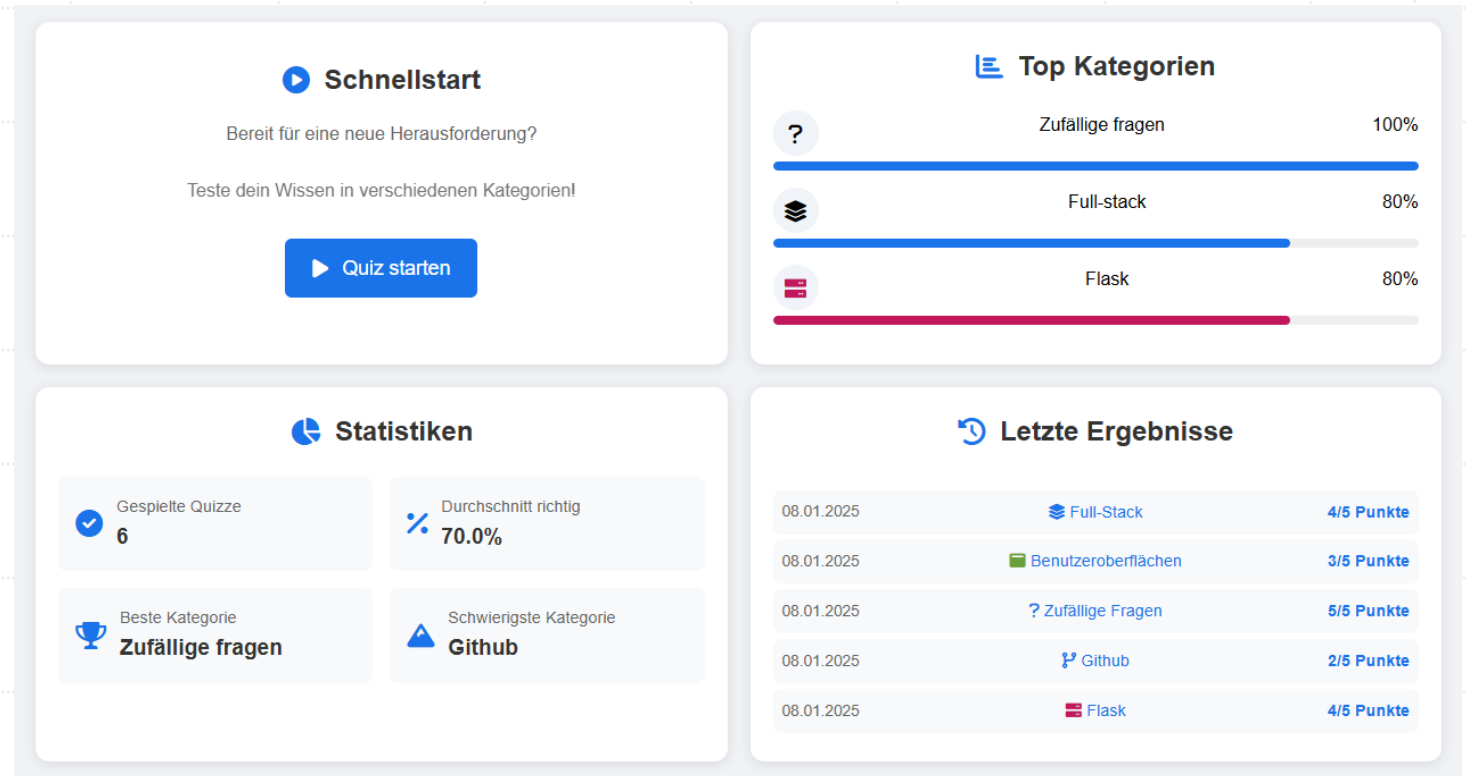
Deine Antwort: Jinja2

[Zum Dashboard](#)[Neues Quiz](#)

2. Quiz-System & Funktionen

Fortschrittssystem:

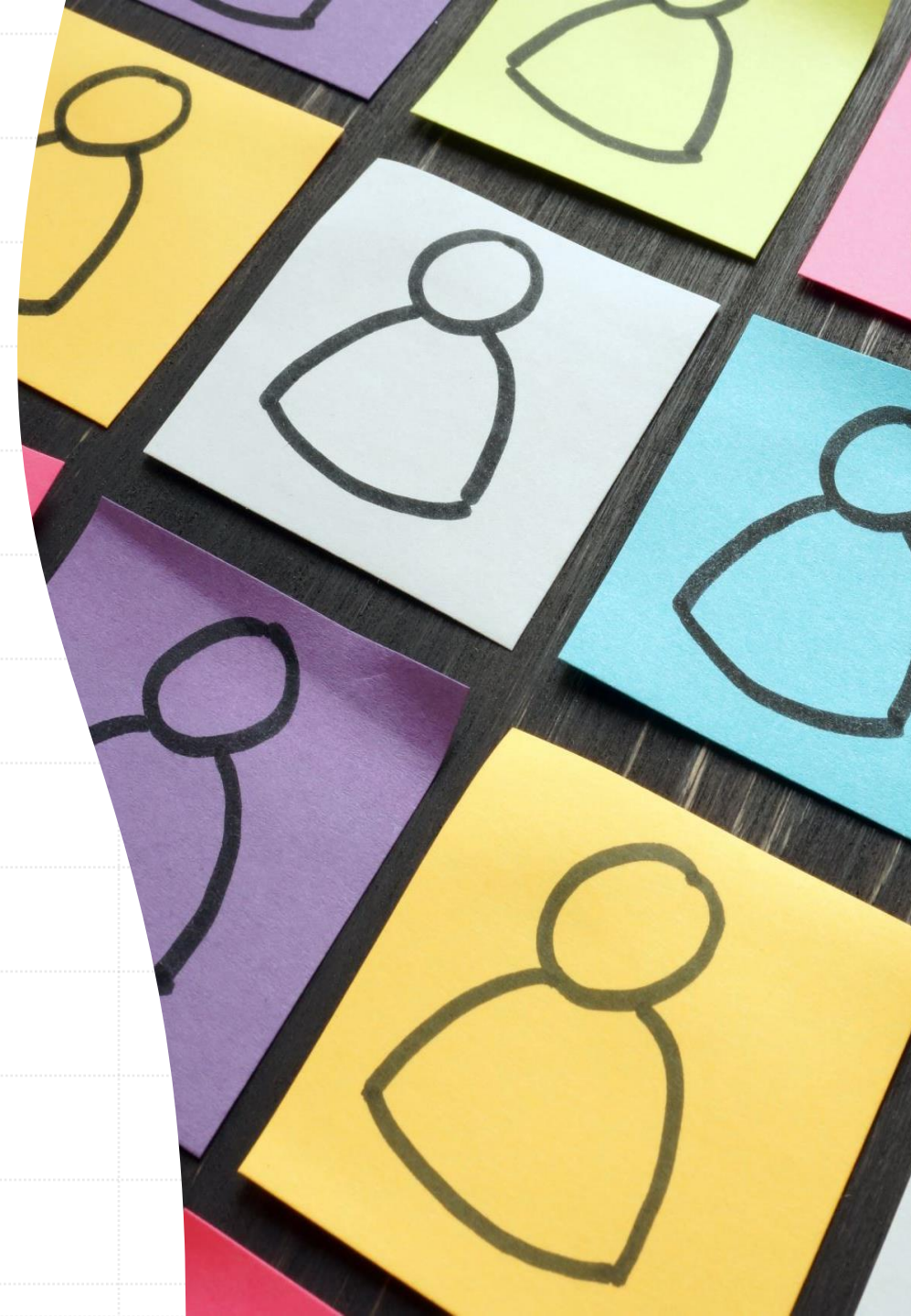
- Transparentes Punktesystem (1 Punkt pro richtige Antwort)
- Detaillierte und dynamische Statistiken:
 - o Gesamtzahl gespielter Quizze
 - o Erfolgsquote
 - o Identifikation von Stärken/Schwächen



2. Quiz-System & Funktionen

Das Freundschaftssystem ermöglicht:

- Suche nach anderen Nutzern
- Freunde hinzufügen/entfernen
- Einsicht in Statistiken der Freunde
- Quiz-Herausforderungen an Freunde



2. Quiz-System & Funktionen

Profil-Sektion:

Statistik-Widget:

- Übersicht gespielter Quizze
- Durchschnittliche Erfolgsquote
- Beste und schwächste Kategorien
- Visualisierung des Lernfortschritts

Benutzerdaten-Management:

- Bearbeitung persönlicher Informationen:
 - o Benutzername
 - o Vor- und Nachname
- Automatische Speicherung der Änderungen

Passwort-Management:

- Sichere Passwortänderung
- Bestätigungsverfahren

Konto-Verwaltung:

- Account-Löschung mit Sicherheitsabfrage



3. Live Demo

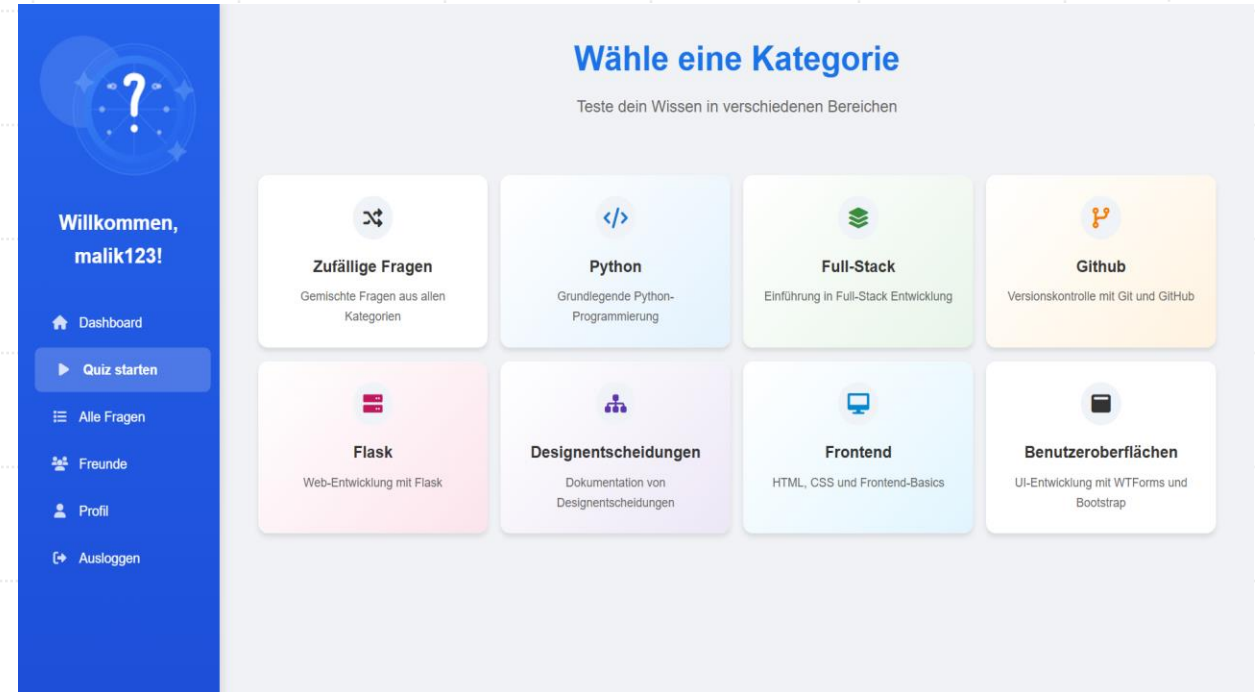
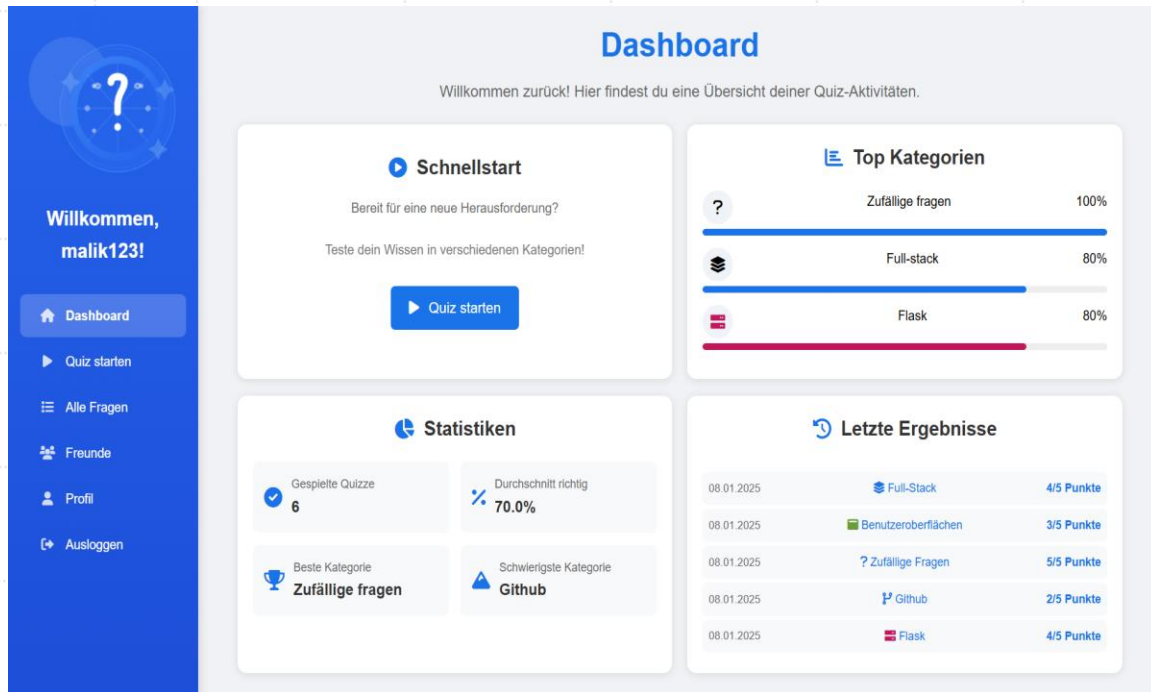
<http://127.0.0.1:5000/>

4. Design

GESTALTUNGSELEMENTE

BENUTZERFREUNDLICHKEIT

MODERNITÄT





5. Technische Architektur

Frontend-Technologien:

- HTML, CSS, JavaScript
 - * JavaScript für interaktive Funktionen
 - * Responsive Design
- Jinja2 Templates
 - * Server-Side Rendering
 - * Template-Vererbung

5. Technische Architektur

Backend Technologien:

Flask

- * Routing und Request-Handling
- * Session-Management
- * Template-Rendering mit Jinja2

SQLite

- * Leichtgewichtige, dateibasierte Datenbank
- * Keine separate Server-Installation nötig

SQLAlchemy

- * ORM für Datenbankoperationen

Sicherheit

- * Session-basierte Authentifizierung
- * Passwort-Hashing mit Werkzeug



6. Zentrale Entwicklungsentscheidungen

SQLite mit SQLAlchemy:

- * Einfache Integration in Flask
- * Keine separate Datenbank-Installation nötig
- * Dateibasiert, ideal für Entwicklung und Testing

Server-Side Rendering

- * Bessere SEO-Optimierung
- * Schnellere initiale Ladezeiten
- * Einfachere Wartung



7. Projektstand & Ausblick

Implementierte Funktionen:

- Komplettes Authentifizierungssystem
- Grundlegende Quiz-Verwaltung
- Dashboard für Lernende
- Responsive Benutzeroberfläche

Geplante Erweiterungen:

- Fragen und Antworten selbst bereitstellen
 - o Personalisierte Quiz-Kategorien
 - o Vorbereitung auf verschiedene Prüfungen
- Medien-Integration (z.B: Bilder)
- Echtzeit-Multiplayer-Modus