

Checkliste Quizly

Bitte erfülle alle Punkte auf dieser Liste, bevor du das Projekt einreichst. **(Definition of Done - DoD)**

1. Technische Anforderungen

Clean Code

- ☒ Funktionen sind maximal 14 Zeilen lang
- ☒ Jede Funktion erfüllt genau eine Aufgabe
- ☒ Alle Funktionsnamen folgen der snake_case Konvention
- ☒ Sprechende Variablennamen sind durchgängig verwendet
- ☒ Alle deklarierten Variablen und Funktionen werden genutzt
- ☒ Auskommentierter Code wurde entfernt

Dokumentation

- ☒ Dokumentation ist vorhanden (Doc String)
- ☒ README.MD Datei existiert und ist aussagekräftig

Django-Spezifisch

- ☒ Code ist in der richtigen Datei
 - ☒ views.py — Nur views, die eine HTTPResponse returnen
 - ☒ functions.py oder utils.py — Neu anlegen für Hilfsfunktionen
- ☒ Das Admin Panel wird gepflegt und lässt das bearbeiten von Quizzes und einzelnen Quizfragen zu

Pythonic Style

- ☒ Code ist [PEP-8](#) compliant
- ☒ Wenn möglich, einhalten

Sonstige Technische Anforderungen

- ☒ Backend und Frontend sind getrennt und kommunizieren über eine **REST-API**
 - ☒ Front-End wird gestellt
 - ☒ Nutze Django im Backend
- ☒ Authentifizierung soll mit **JWT** und **HTTP-ONLY-COOKIES** eingerichtet werden
- ☒ Du benötigst unbedingt **FFMPEG** global installiert (dies benötigt Whisper AI). Bitte unbedingt auch in deiner README mit angeben, dass dies eben benötigt wird.
- ☒ Es sollen nur YouTube Videos in Audio umgewandelt werden. Nutze hierfür **yt_dlp** (<https://github.com/yt-dlp/yt-dlp/tree/2025.07.21>)
- ☒ Nutze für das Transkribieren der Audio Files Whisper AI (<https://github.com/openai/whisper>). Das ganze kann lokal verwendet werden.
- ☒ Um ein Quiz zu erstellen, nutze die **KI Gemini Flash**. Die Verwendung dieser Flash Variante ist kostenlos. Du benötigst lediglich einen API Key (<https://github.com/googleapis/python-genai>)

2. Funktionale Anforderungen - Benutzeraccount & Registrierung:

User Story 1: Benutzerregistrierung

Als neuer Benutzer möchte ich mich bei Quizly registrieren können, um Zugang zur Plattform zu erhalten und Inhalte anzusehen.

- ☒ Es gibt ein Registrierungsformular mit Feldern für E-Mail, Passwort und Passwortbestätigung.

Das Backend sollte nach Validierung den User speichern.

- ☒ Bei ungültiger Eingabe (z.B. bereits verwendete E-Mail) erhält der Benutzer eine Fehlermeldung.
- ☒ Ist man bereits registriert, kann man zum Anmeldeformular wechseln.

User Story 2: Benutzeranmeldung

Als registrierter Benutzer möchte ich mich bei Quizly anmelden können, um auf mein Konto zuzugreifen und Inhalte anzusehen.

- ☒ Es gibt ein Login-Formular mit Feldern für E-Mail und Passwort.
- ☒ Bei falscher Eingabe erhält der Benutzer eine Fehlermeldung.
- ☒ Fehlermeldungen sind aus Sicherheitsgründen allgemein gehalten.
- ☒ **Das Backend soll bei einem Login den User validieren. Bei gültiger Validierung soll durch JWT je ein "ACCESS-TOKEN" als auch ein "REFRESH-TOKEN" erstellt werden. Diese werden dann per HTTP-ONLY-COOKIES an das FrontEnd geschickt**

~~und in den Cookies gespeichert. Diese Cookies dienen dann zur weiteren Authentifizierung.~~

- ☒ ~~Nach erfolgreicher Anmeldung wird der Benutzer zur Startseite weitergeleitet. Auf der Startseite kann der User entweder ein neues YouTube Video hochladen oder sich seine letzten Quizze ansehen.~~
- ☒ ~~Sollte der Nutzer noch kein Konto haben, kann er zum Registrierungsformular wechseln.~~

User Story 3: Benutzerabmeldung

Als Benutzer möchte ich mich von Quizly abmelden können, damit niemand ohne meine Zustimmung auf meinen Account zugreifen kann.

- ☒ ~~Es gibt eine "Logout"-Option in der Benutzeroberfläche.~~
- ☒ ~~Nach Auswahl dieser Option werde ich sicher aus der Anwendung ausgeloggt und zum Login-Bildschirm weitergeleitet.~~
 - **Bei einem Logout wird auf entsprechende API im Backend zugegriffen. Da die Authentifizierung dann über die Token in den Cookies stattfindet, kann dies dem entsprechenden User zugeordnet werden und die Cookies im FrontEnd werden entsprechend gelöscht.**
- ☒ ~~Nach dem Abmelden sind meine persönlichen Daten und Einstellungen ohne erneutes Einloggen nicht zugänglich.~~
 - **Es werden im Backend die bereits benutzten Tokens nach dem Logout des Users unbrauchbar und kommen auf eine "Blacklist". Diese können dann nicht mehr verwendet werden. Der User muss sich somit immer neu anmelden und kann die alten Tokens nicht mehr verwenden.**

3. Funktionale Anforderungen - Quiz

User Story 4: Quizübersicht - neues Quiz generieren

Als angemeldeter habe ich die Möglichkeit über ein Inputfeld eine YouTube URL einzugeben und mir ein neues Quiz erstellen zu lassen.

- ☒ ~~Das Dashboard zeigt ein Input-Feld für die Eingabe einer URL. Hier kann ein neues Quiz erstellt werden.~~
 - **Das Backend erhält eine URL. Aus dieser wird dann entsprechend das Video in ein Audiofile mit FFMPEG umgewandelt. Dieses Audio-File wird dann im Anschluss mit Whisper AI transkribiert. Das Transkript wird dann entsprechend für ein Prompt aufbereitet damit daraus dann mit der Gemini-Flash AI ein Quiz mit je 10 Fragen und 4 Antwortmöglichkeiten erstellt werden kann. Dieses Quiz wird dann gespeichert.**

User Story 5: Quizübersicht - bereits gespielte Quizze erneut spielen

Als angemeldeter Benutzer möchte ich eine Übersicht über meine bereits erstellten Quizze sehen, um diese noch einmal zu durchlaufen zu können.

- ☒ ~~Dem User wird eine Übersicht seiner letzten Quizze angezeigt. Über einen Klick auf das entsprechende Quiz kann dann in die Detailansicht des Quizzes zugreifen und hieraus das Quiz erneut starten.~~

User Story 6: Quiz-Detailansicht

Als Benutzer möchte ich meine Quizze verwalten. In dieser Übersicht kann ich das von der KI bereits vordefinierte Feld TITEL und BESCHREIBUNG ändern. Ich sehe zusätzlich in der Übersicht ein Embedded Video des verwendeten YouTube Videos .

- ☒ ~~Es kann der Titel und die Beschreibung des Quiz geändert werden.~~
 - **Alle Änderungen am Titel und der Beschreibung werden im Backend gespeichert.**
- ☒ ~~Ich sehe das verwendete YouTube Video und kann es mir direkt ansehen.~~
 - **Das YouTube Video wird als URL in der Datenbank gespeichert. Über diese URL wird es dann im FrontEnd als Embedded Video eingebunden.**
- ☒ ~~Über Buttons kann das Quiz gestartet werden.~~

User Story 7: Quiz-Detailansicht Sidebar

Als Benutzer habe ich die Möglichkeit über eine Sidebar auf meine Heute erstellten Quizze zuzugreifen. Zudem erhalte ich eine Übersicht über die erstellten Quizze der letzten 7 Tage.

- ☒ ~~Es gibt eine Übersicht der **Heute** erstellten Quizze bzw. der **letzten 7 Tage**. Mit Klick auf den Titel kann ich wiederum auf entsprechendes Quiz in die Detailansicht wechseln. Zudem gibt es die Möglichkeit auf alle Quizze zuzugreifen. Hier können dann auch einzelne Quizze gelöscht werden.~~

User Story 8: Quiz

Als Benutzer möchte ich das erstellte Quiz spielen. Hierzu erhalte ich insgesamt 10 Fragen mit je 4 Auswahlmöglichkeiten.

- ☒ ~~Der Spielfortschritt wird automatisch gespeichert.~~
- ☒ ~~Ich kann zu bereits beantworteten Fragen zurück gehen und die Auswahl ändern.~~

User Story 9: Quiz Auswertung

Als Benutzer erhalte ich am Ende des Quizzes eine Auswertung über meine richtigen Fragen mit einer prozentualen Quote. Ich habe nun die Auswahlmöglichkeit das Quiz zu wiederholen oder mir die Antworten anzeigen zu lassen.

- ☒ Ich kann mir die Antworten anzeigen lassen. Optional auch den direkten Vergleich meiner Antworten und die tatsächliche Lösung.
- ☒ Es ist ebenfalls ein Button vorhanden in dem ich zur Quizübersicht gelange um ein neues Quiz generieren zu lassen oder ein altes erneut zu spielen.

User Story 10: Rechtliche Informationen

Als Benutzer möchte ich Zugang zu rechtlichen Informationen wie Datenschutzerklärung und Impressum haben, um mich über meine Rechte und die Nutzungsbedingungen zu informieren.

- ☒ Es gibt leicht zugängliche Links zur Datenschutzerklärung und zum Impressum im Footer der Website. **Die Daten des jeweiligen Betreibers müssen noch entsprechend ergänzt werden und auf seine Daten aktualisiert werden.**
- ☒ Die Informationen sind klar strukturiert und in verständlicher Sprache verfasst.
- ☒ Die Seiten sind responsiv und auf allen Geräten gut lesbar.