

1. Zielbestimmung

1.1 Projektziel:

Entwicklung einer webbasierten Plattform zur Verwaltung, Kategorisierung und Nutzung von schulischen Lernmaterialien (z. B. PDF, Markdown, Textdateien). Nutzer sollen Materialien hochladen, mit Tags versehen, durchsuchen und mit einer KI interagieren können, um Inhalte besser zu verstehen oder zusammenfassen zu lassen.

1.2 Zielgruppe:

- Schüler:innen
 - Lehrkräfte
 - Bildungseinrichtungen
 - Studierende
-

2. Produkteinsatz

2.1 Einsatzbereich:

- Browserbasierte Webanwendung (Desktop und Mobile)
- Später optional: Mobile App (Android/iOS)

2.2 Betriebssysteme & Endgeräte:

- Plattformunabhängig (Chrome, Firefox, Safari, Edge)
 - Responsive Design für mobile Geräte
-

3. Funktionale Anforderungen

3.1 Benutzerverwaltung

- Registrierung / Login (E-Mail/Name + Passwort)
- Rollen: Admin, (Lehrer?), Schüler
- Profilbearbeitung
- Passwort-Reset
- Bestätigungsemail?
- 2FA?

3.2 Datei-Upload & -Verwaltung

- Upload von PDF, .md, .txt Dateien (andere Dateien?)
- Drag & Drop Upload
- (Dateigröße begrenzt)
- Kategorisierung mit Tags (z. B. „1. Klasse“, „Syt“, „Elektrotechnik“)

- Organisation in Datenbank
- Datei-Vorschau (PDF, Markdown-Renderer, Textansicht)
- Versionierung von Dateien

3.3 KI-gestützte Funktionen

- Chat mit KI zu einer ausgewählten Datei
- Automatische Zusammenfassungen
- Generierung von Quizfragen
- Lernkarten-Erstellung
- Erklärung von Fachbegriffen / Vereinfachte Sprache
- Themenvorschläge / Weiterführende Inhalte

3.4 Suche & Filter

- Volltextsuche in Dokumenten
- Filter nach Tags, Dateityp, Klassenstufe, Kurs, Fach, ...
- Favoriten / Merkliste
- Häufig genutzte Dateien

3.5 Kollaboration & Notizen

- Kommentarfunktion pro Datei
- Notizen (persönlich & öffentlich)
- Freigabe von Dateien mit Rechten (Lesen, Bearbeiten)
- Gemeinsame Ordner für Gruppen / Klassen

4. Nicht-funktionale Anforderungen

4.1 Sicherheit & Datenschutz

- (DSGVO-Konformität)
- SSL-Verschlüsselung
- Rechteverwaltung
- Datenverschlüsselung im Ruhezustand und bei Übertragung
- Keine Weitergabe von Nutzerdaten an Dritte

4.2 Systemleistung

- Antwortzeiten minimieren
- Skalierbarkeit

4.3 Barrierefreiheit

- Tastaturnavigation
 - Kontraste und Schriftgrößen
-

5. Technische Rahmenbedingungen

- **Frontend:** React, TailwindCSS
 - **Backend:** ASP.NET Core Web API (C#)
 - **Datenbank:** PostgreSQL
 - **KI-Integration:** Anbindung an externe KI-APIs (z. B. OpenAI GPT-4, Azure OpenAI) über HTTP-Clients oder Model lokal laufen lassen **ABER selbst trainieren bevorzugt**
 - **(Authentifizierung:** JWT, OAuth2 (später SSO möglich))
-

6. Benutzeroberfläche (UI/UX)

- Modernes, responsives Design
 - Übersichtliches Dashboard mit Zugriff auf zuletzt verwendete Dateien, Favoriten (+Upload-Statistiken?)
 - Dokumentenübersicht mit:
 - Sortier- & Filteroptionen (nach Tags, Typ, Kurs, Klasse usw.)
 - Vorschau (PDF-Viewer, Markdown-Renderer, Textansicht)
 - Schnellaktionen (z. B. „Teilen“, „Bearbeiten“, „Löschen“)
 - Chat-Interface mit KI (ähnlich wie ChatGPT)
 - Weitere KI-Funktionen (Lernkarten erstellen, Zusammenfassung / Erklärung, Quiz-Erstellung, ...)
 - Upload-Dialog mit Tageingabe
 - Tag-Verwaltung & Tag-Vorschläge (z. B. automatische Vorschläge nach Inhalt)
 - Dateikommentare & persönliche Notizen
 - Favoriten & Merkliste
 - Benutzerprofil & Einstellungen
-

8. Offene Punkte / Erweiterungen (optional)

- Mobile App
- Gamification (Badges, XP)
- Kalender-Integration
- Export-Funktion (PDF, CSV von Lerninhalten)