

# Pflichtenheft LernApp

Jannis Nau

Florian Kaiser

Jochen Heimüller

## 1. Grundlegende Problemstellung

Physische Karteikarten sind oft mühsam zu Transportieren und nichtmehr zeitgemäß. Digitale Lern-Apps sind auf dem Vormarsch und bieten dank intelligenter Algorithmen verbesserten Lernerfolg. Auch Hochschulalltag fällt es Studierenden häufig schwer, sich Wissen langfristig und effizient anzueignen. Klassische Lernmethoden wie das wiederholte Lesen von Skripten führen nur zu kurzfristigen Lernerfolgen. Digitale Lernkarten-Systeme (z. B. Anki) setzen auf sogenannte *Spaced Repetition*-Algorithmen, um Wissen durch zeitlich optimierte Wiederholungen zu festigen.

**Ziel** des Projekts Lern-App ist die Entwicklung einer **einfach bedienbaren, webbasierten Lernkarten-App**, die auf modernen Softwarearchitekturprinzipien basiert. Der Fokus liegt auf:

- einer klar strukturierten Architektur mit Trennung von Frontend, Backend und Datenhaltung,
- einer intuitiven Benutzeroberfläche,
- der Umsetzung eines anpassbaren Wiederholungsalgorithmus.

Das System soll prototypisch zeigen, wie Lernprozesse technisch und architektonisch effizient unterstützt werden können.

## 2. Technologieauswahl

Komponente	Technologie	Begründung
Frontend	<b>Angular (TypeScript)</b>	Klare Komponentenstruktur, gute Unterstützung für modulare und reaktive Webanwendungen, etablierter Enterprise-Standard.
Backend	<b>.NET 8 / ASP.NET Core Web API (C#)</b>	Hohe Performance, saubere Trennung von Logik und Datenzugriff, gute Unterstützung für REST-basierte Architekturen.
Datenbank	<b>SQLite (lokal) oder PostgreSQL (optional)</b>	Relationales Schema für Karten/Decks; SQLite genügt für Prototyp.
Styling	<b>Angular Material oder Bootstrap</b>	Einheitliches UI-Design, responsive Darstellung.
Architektur	<b>Client-Server mit REST-Schnittstelle (MVC / Clean Architecture)</b>	Fördert Wartbarkeit, Testbarkeit und Erweiterbarkeit.
Tests	<b>xUnit / NUnit (Backend), Jasmine + Karma (Frontend)</b>	Sicherstellung der Funktionalität und Stabilität.

## 3. Use Cases

Titel	Beschreibung
<b>Karte anlegen</b>	Benutzer erstellt eine Karte mit Frage, Antwort und optionalen Tags.
<b>Deck verwalten</b>	Erstellen, Umbenennen, Löschen und Gruppieren von Kartendecks.

Titel	Beschreibung
<b>Lernmodus starten</b>	Benutzer startet eine Lernsession für ein Deck. Karten werden je nach Lernfortschritt angezeigt.
<b>Karte bewerten</b>	Nach dem Anzeigen der Antwort bewertet der Benutzer, wie gut er die Karte wusste (“einfach”, “mittel”, “schwer”).
<b>Statistik anzeigen</b>	Anzeige des Lernfortschritts und Wiederholungsstatus.

## 4. Muss- / Kann-Kriterien

### Muss-Kriterien

- Erstellung und Verwaltung von Lernkarten und -decks
- Lernmodus
- Persistente Datenspeicherung
- Weboberfläche
- REST-basierte Kommunikation zwischen Frontend und Backend
- Grundlegende Testabdeckung

### Kann-Kriterien

- Benutzerverwaltung
- Import/Export von Decks
- Fortschrittsdiagramme
- Optionale Cloud-Synchronisierung