

Projektdokumentation		
Projektname	Lern- und Prüf-Simulator	
Dokumententitel	Pflichtenheft	
Erstellt von	Mo / Eduard	
Dokument-Version	1.0	Erstellt am 18.08.2025

Inhaltsverzeichnis

1. Ausgangssituation und Zielsetzung	2
1.2 Zielbestimmung	3
2. Produkteinsatz	4
2.1 Zielgruppen	4
2.2 Einsatzumgebung/Plattformen	4
2.3 Sprachunterstützung	4
2.4 Nutzungsszenarien	4
3. Funktionale Anforderungen	5
3.1 Muss-Kriterien	5
3.2 Kann-Kriterien	6
3.3 Abgrenzungskriterien	6
4. Nicht-funktionale Anforderungen	6
4.1 Zugriff auf die App	6
4.2 Design und UX	7
4.3 Datenvolumen	8
4.4 Datenmodell	9
5. Lieferumfang	9
5.1 Zu übergebende Dokumente	9
5.2 Bereitstellung der Software	10
5.3 Schulungsleistungen	10
5.4 Web-API	10
6. Abnahmekriterien	11
6.1 Qualitätsziele	11
6.2 Testszenarien	11

1. Ausgangssituation und Zielsetzung

1.1 Ausgangssituation

Die BITLC GmbH bietet verkürzte Ausbildungen zum Fachinformatiker in den Fachrichtungen Anwendungsentwicklung und Systemintegration an. Ein zentraler Bestandteil dieser Ausbildung ist die Vorbereitung auf die LPIC-1-Zertifizierung. Für diesen Zweck wird derzeit eine Software eingesetzt, die ausschließlich Prüfungssimulationen anbietet. Dabei müssen Lernende Fragen unter Zeitdruck beantworten. Dieses Vorgehen hat den Nachteil, dass Lernende häufig ohne ausreichendes Fachverständnis zum Raten neigen, was zur Folge hat, dass sich falsche Antworten im Gedächtnis verankern. Damit ist der Lerneffekt begrenzt und teilweise sogar kontraproduktiv.

Darüber hinaus zeigt die bisherige Software weitere Einschränkungen:

- **Fehlender Lernmodus:** Es gibt keine Möglichkeit, Fragen zunächst ohne Zeitlimit zu üben und die Lösung kontrolliert aufzudecken.
- **Fehlende Flexibilität:** Inhalte sind direkt in der Software eingebettet und lassen sich nur mit großem Aufwand austauschen oder erweitern.
- **Eingeschränkte Didaktik:** Es existiert keine klare Trennung zwischen *Lernen* und *Prüfen*.
- **Begrenzte Plattformunterstützung:** Zwar läuft die Software auf gängigen Systemen, jedoch fehlt eine konsequent webbasierte Lösung, die auf allen Endgeräten mit Browser lauffähig ist.

Betroffen von diesen Nachteilen sind mehrere Zielgruppen: Auszubildende und Studierende, die sich auf Prüfungen vorbereiten, Dozente, die zuverlässige Werkzeuge für Unterricht und Prüfungsvorbereitung benötigen, sowie die Ausbildungscoordination, die auf eine flexible und zukunftssichere Lösung angewiesen ist.

Aus diesen Gründen besteht der Wunsch, eine neue Softwarelösung zu schaffen, die über einen reinen Prüfungsmodus hinausgeht und den Lernenden einen nachhaltigen und flexiblen Wissensaufbau ermöglicht.

1.2 Zielbestimmung

Das Ziel des Projekts ist die Entwicklung einer webbasierten Anwendung, die sowohl individuelles Lernen als auch die realistische Simulation von Prüfungssituationen unterstützt. Die neue Lösung soll folgende Anforderungen erfüllen:

- Lernmodus: Fragen können ohne Zeitlimit durchgearbeitet werden. Die Lernenden entscheiden selbst, wann sie die Lösung aufdecken und ob sie eine Antwort abgeben. Sofortiges Feedback sowie Erklärungen sollen zur Verfügung stehen.
- Prüfungssimulation: Lernende können eine realistische Prüfung mit Zeitvorgaben und einer definierten Anzahl von Fragen durchführen. Während der Bearbeitung wird kein Feedback gegeben; am Ende wird eine detaillierte Ergebnisübersicht angezeigt.
- Fragetypen: Unterstützung der gängigen Typen Single-Choice, Multiple-Choice und Fill-in (Freitexteingaben).
- Flexible Inhalte: Fragen und Kataloge werden über eine Web-API bereitgestellt, sodass die App unabhängig von den Inhalten bleibt und diese leicht ausgetauscht oder erweitert werden können.
- Plattformunabhängigkeit: Die App soll auf allen modernen Browsern und Endgeräten (PC, Laptop, Tablet, Smartphone) lauffähig sein.
- Benutzerfreundlichkeit: Intuitive Bedienung mit klarer Navigation, responsivem Design und konsistenter Nutzerführung.

Rahmenbedingungen:

In der ersten Ausbaustufe wird es keine Authentifizierung geben, es werden keine personenbezogenen Daten gespeichert und keine Simulationsergebnisse dauerhaft abgelegt. Eine Internationalisierung ist vorerst nicht vorgesehen; die App wird zunächst nur auf Deutsch verfügbar sein.

Erfolgskriterien:

- Die App bietet zwei klar getrennte Modi: Lernen und Simulation.
- Alle im Lastenheft definierten Fragetypen werden unterstützt.
- Die Inhalte werden vollständig über eine Web-API bereitgestellt.
- Die App ist plattformunabhängig auf den gängigen Browsern nutzbar.
- Lernende können sowohl im Lernmodus als auch in der Simulation ihr Wissen effektiv und nachhaltig aufbauen.

2. Produkteinsatz

2.1 Zielgruppen

Die Hauptzielgruppe sind Lernende (Auszubildende, Umschüler:innen, Studierende), die sich auf die LPIC-1-Prüfung oder vergleichbare IT-Zertifizierungen vorbereiten. Sie benötigen eine flexible, leicht zugängliche Lösung, die sowohl individuelles Üben als auch realistische Prüfungssimulationen erlaubt.

Weitere Zielgruppen sind Dozent:innen und Ausbilder:innen, die das Tool im Unterricht einsetzen, sowie die Ausbildungscoordination, die auf eine zukunftssichere, leicht erweiterbare Lösung angewiesen ist.

Indirekt profitieren auch Bildungsträger und Prüfungseinrichtungen, da die Lernenden besser vorbereitet und motivierter sind.

2.2 Einsatzumgebung/Plattformen

Die Anwendung ist plattformunabhängig und funktioniert auf allen modernen Endgeräten mit Webbrowser. Unterstützt werden:

- Desktop-PCs und Laptops mit gängigen Betriebssystemen (Windows, macOS, Linux)
- Tablets (iOS, Android) ab 7 Zoll Bildschirmgröße
- Smartphones (iOS, Android) ab einer Displaybreite von 360 Pixel

Technische Anforderungen:

- Aktuelle Browser (letzte zwei Hauptversionen von Chrome, Firefox, Edge, Safari)
- Stabile Internetverbindung ab 3G; empfohlen wird mindestens 10 Mbit/s Download
- Serverseitig: Hosting auf Standard-Webserver oder Cloud-Umgebung, keine besonderen Hardwareanforderungen

2.3 Sprachunterstützung

Die Anwendung wird zunächst einsprachig in Deutsch bereitgestellt. Damit wird die Hauptzielgruppe (Auszubildende in Deutschland) vollständig abgedeckt.

Eine spätere Erweiterung auf weitere Sprachen (z. B. Englisch) ist möglich, da die technische Architektur (Frontend mit externen Textressourcen) eine Internationalisierung (i18n) vorsieht.

2.4 Nutzungsszenarien

Die Software wird in folgenden Szenarien eingesetzt:

- Selbststudium zuhause: Lernende nutzen die App eigenständig, um Themen gezielt zu wiederholen, Fragenkataloge durchzugehen und im Lernmodus Wissen aufzubauen.
- Unterricht im Klassenraum: Dozent:innen empfehlen den Einsatz als Ergänzung zum Unterricht. Lernende können bestimmte Kataloge bearbeiten und anschließend Simulationen durchführen.
- Prüfungsvorbereitungskurse: Vor Prüfungen wird die App eingesetzt, um unter realistischen Bedingungen (Simulation) den Wissensstand zu überprüfen.
- Unterwegs/Mobilnutzung: Dank responsivem Design können Lernende auch unterwegs (z. B. mit Smartphone oder Tablet) kurze Lerneinheiten durchführen.

3. Funktionale Anforderungen

3.1 Muss-Kriterien

Die folgenden Anforderungen sind zwingend umzusetzen, um die Kernfunktionalität der Anwendung zu gewährleisten:

- Zwei Betriebsmodi: Lernmodus und Prüfungssimulation, klar voneinander getrennt.
- Fragetypen: Unterstützung von Single-Choice, Multiple-Choice und Fill-in.
- Navigation: Vor- und Zurücknavigation zwischen Fragen, Rücksprung zur Katalogübersicht.
- Feedback: Im Lernmodus sofortiges Feedback (richtig/falsch) und Anzeige von Lösungen sowie Erklärungen.
- Simulation: In der Simulation kein Feedback während der Bearbeitung; Ergebnisübersicht am Ende mit Gesamtbewertung und Auflistung falscher Antworten.
- Datenbereitstellung: Alle Fragen und Kataloge werden über eine externe Web-API bezogen.
- Plattformunabhängigkeit: Funktioniert auf allen modernen Browsern (Chrome, Firefox, Edge, Safari).

3.2 Kann-Kriterien

Diese Anforderungen sind wünschenswert, erhöhen den Nutzen und Komfort, sind jedoch nicht zwingend für die erste Version:

- Zufällige Reihenfolge: Fragen und Antwortoptionen können zufällig gemischt werden.
- Teilfeedback: Möglichkeit, auch in der Simulation optional bestimmte Hilfen oder Zwischenfeedback zu aktivieren.
- Themenauswahl: Lernende können bestimmte Themen oder Unterkataloge gezielt auswählen.
- Timer-Konfiguration: In der Simulation kann die Zeitvorgabe angepasst werden (z. B. 30 Sekunden bis 2 Minuten pro Frage).
- Barrierefreiheit: Erweiterte Unterstützung für Screenreader, hohe Kontraste, Tastaturbedienung.

3.3 Abgrenzungskriterien

Die folgenden Punkte sind explizit nicht Bestandteil der ersten Ausbaustufe:

- Keine Authentifizierung: Es werden keine Benutzerkonten angelegt und keine Login-Funktion implementiert.
- Keine Speicherung von Ergebnissen: Bearbeitete Fragen und Prüfungsergebnisse werden nicht persistent gespeichert.
- Keine Statistiken: Es gibt keine Auswertungen über Lernfortschritte oder Vergleiche zwischen Lernenden.
- Kein Offline-Modus: Ständige Internetverbindung erforderlich.
- Keine Integration in Fremdsysteme: Kein Export von Daten an externe Plattformen oder Learning-Management-Systeme in v1.

4. Nicht-funktionale Anforderungen

4.1 Zugriff auf die App

Die Anwendung muss niedrigschwellig und ohne Hürden zugänglich sein. Der Zugriff erfolgt ausschließlich über einen Webbrowser, ohne lokale Installation oder besondere Systemrechte. Damit ist sichergestellt, dass Lernende die App von jedem gängigen Endgerät aus nutzen können (PC, Laptop, Tablet, Smartphone).

Verfügbarkeit & Erreichbarkeit: Die App soll rund um die Uhr erreichbar sein. Kurzzeitige Wartungsfenster sind außerhalb üblicher Unterrichtszeiten zu planen und rechtzeitig anzukündigen. Ein stabiler Zugriff ist auch bei mittleren Bandbreiten (z. B. 4G-Mobilfunk) zu gewährleisten, sodass Lernende die App unterwegs nutzen können.

Kompatibilität: Unterstützt werden die letzten zwei Hauptversionen der gängigen Browser (Chrome, Firefox, Edge, Safari). Die Darstellung und Bedienung sind auf verschiedenen Bildschirmgrößen konsistent, sodass keine Funktionsverluste auf kleineren Displays auftreten.

Sicherheit & Datenschutz (rahmengebend): In der Anfangsversion werden keine personenbezogenen Daten erhoben oder gespeichert. Die Kommunikation zwischen Client und Inhalte-Quelle (Web-API) soll verschlüsselt erfolgen (z. B. via HTTPS), um die Inhalte vor unbefugtem Mitlesen zu schützen.

Barrierearme Nutzung: Die App bietet eine Tastaturbedienung, klare Fokus-Indikatoren sowie ausreichende Kontraste, sodass auch Nutzende mit Einschränkungen die wichtigsten Funktionen problemlos bedienen können.

4.2 Design und UX

Die Oberfläche folgt dem Prinzip „einfach, klar, fokussiert“. Ziel ist eine intuitive Nutzerführung, die Lernenden schnelle Orientierung und fehlerarme Bedienung ermöglicht.

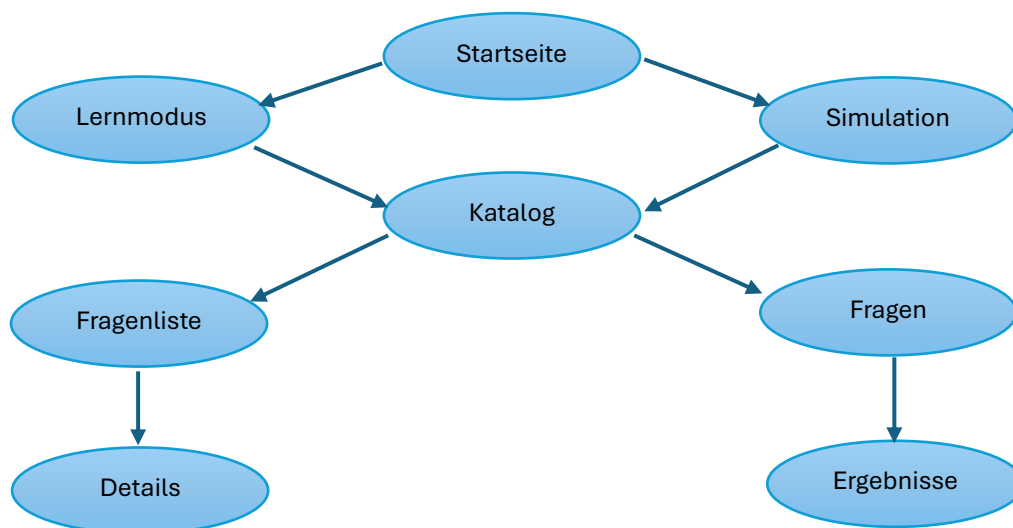
Grundprinzipien:

- Klare Navigationsstruktur: Start → Thema → Katalog → Fragenliste → Frage-Detail. Die Nutzer erkennen jederzeit, wo sie sich befinden (z. B. durch Breadcrumbs) und welche Optionen als Nächstes sinnvoll sind (z. B. „Nächste Frage“, „Zurück zur Liste“).
- Trennung der Modi: Der Lernmodus (Feedback nach Wunsch, Lösung on-demand) ist optisch und inhaltlich klar vom Simulationsmodus getrennt (kein Zwischenfeedback, Timer, Ergebnisübersicht).
- Konsistente Interaktionselemente: Schaltflächen, Links und Zustandsanzeigen (Laden, Fehler, Timer) sind in Bezeichnung, Position und Verhalten einheitlich.
- Responsives Layout: Inhalte passen sich unterschiedlichen Bildschirmgrößen an, ohne an Verständlichkeit zu verlieren; Bedienelemente bleiben auch per Touch gut erreichbar.
- Lesbarkeit: Ausreichend weißer Raum, verständliche Sprache, klare Typografie und angemessene Kontraste unterstützen längere Lerneinheiten ohne Ermüdung.

Spezifische UX-Anforderungen:

- Lernmodus: Pro Frage steht eine eindeutige Aktion „Lösung anzeigen“ zur Verfügung. Nach dem Anzeigen ist unverwechselbar erkennbar, was die richtige Antwort wäre; optional sind kurze Erklärungen hinterlegt.
- Simulation: Während der Bearbeitung wird kein Feedback gezeigt; der Timer ist sichtbar, aber dezent, um nicht unnötig zu stressen. Die Ergebnisübersicht am Ende ist leicht verständlich (Score, Übersicht falsch beantworteter Fragen).
- Fehler- und Leersituationen: Bei verlorener Verbindung oder leeren Katalogen erscheinen klare Hinweise mit konkreter Handlungsoption (z. B. „Erneut versuchen“, „Zur Katalogauswahl“).
- Hilfetexte: Kurze, kontextsensitive Hinweise unterstützen ungeübte Nutzer, ohne zu überfrachten.

Sitemap - Lern- und Prüf-Simulator



4.3 Datenvolumen

Das System muss mit variierenden Mengen an Fragen umgehen können, ohne die Nutzbarkeit zu beeinträchtigen.

Ausgangslage: Der initiale Bestand umfasst 12 Kataloge mit jeweils ca. 85–120 Fragen.

Zielvorgaben:

- Das Laden eines vollständigen Katalogs soll üblicherweise unter 5 Sekunden dauern (inklusive Anzeige), wenn eine übliche Mobilfunkverbindung (4G) genutzt wird.
- Seiten- und Zustandswechsel (z. B. „Nächste Frage“) sollen flüssig und ohne merkliche Verzögerung erfolgen.
- Die App soll skalierbar sein, sodass das spätere Hinzufügen weiterer Kataloge/Fragen ohne strukturelle Änderungen möglich ist.

Datenqualität: Fragen und Antworten werden einheitlich strukturiert (ein konsistentes Format), sodass sie von der App zuverlässig interpretiert werden können. Inhalte sind sprachlich eindeutig formuliert, frei von Tippfehlern und fachlich korrekt.

4.4 Datenmodell

Die Fragen und Antworten werden in einem einheitlichen JSON-Format bereitgestellt. Jede Frage enthält eine eindeutige ID, den Fragetext, mehrere Antwortoptionen (mit Markierung der korrekten Antwort/en) sowie optional eine Erklärung.

```
1 Beispiel (vereinfacht):  
2 {  
3   "id": 101,  
4   "frage": "Welcher Befehl zeigt die aktuelle IP-Adresse an?",  
5   "antworten": [  
6     {"id": 1, "text": "ifconfig", "korrekt": true},  
7     {"id": 2, "text": "ipconfig", "korrekt": false},  
8     {"id": 3, "text": "ping", "korrekt": false}  
9   ],  
10  "erklaerung": "Unter Linux zeigt 'ifconfig' die Netzwerkkonfiguration an."  
11 }
```

Dieses Modell stellt sicher, dass die App zuverlässig Fragen und Antworten interpretieren kann und erleichtert eine spätere Erweiterung um weitere Metadaten (z. B. Schwierigkeitsgrad, Themenbereich).

5. Lieferumfang

5.1 Zu übergebende Dokumente

Im Rahmen der Projektergebnisse werden folgende Dokumente zur Verfügung gestellt:

- Systemdokumentation: Beschreibung der Architektur, Datenstrukturen und Schnittstellen. Enthält auch eine Übersicht der technischen Rahmenbedingungen und Anleitungen für Erweiterungen.

- Benutzerhandbuch: Einfache, verständliche Anleitung für Lernende und Dozenten. Beinhaltet eine Einführung in die Nutzung der Modi (Lernmodus, Simulation) sowie häufig gestellte Fragen (FAQ).
- API-Spezifikation: Technische Dokumentation der Web-API, mit der die Fragenkataloge ausgeliefert werden. Enthält die Definition der Endpunkte, Parameter und Antwortstrukturen.

5.2 Bereitstellung der Software

Die Bereitstellung der Software erfolgt als webbasierte Anwendung. Dies umfasst:

- Frontend: Bereitstellung als Angular-Projekt (kompiliert für Browsernutzung). Zugriff erfolgt über eine URL, keine lokale Installation nötig.
- Backend: Bereitstellung einer REST-API mit Zugriff auf die Fragenkataloge (JSON-Dateien). Hosting auf Standard-Server oder Cloud-Umgebung.
- Deployments: Einrichtung einer lauffähigen Instanz (Produktivsystem) sowie optional einer Testumgebung für interne Qualitätssicherung.

5.3 Schulungsleistungen

Da die Anwendung intuitiv und leicht verständlich konzipiert ist, sind nur minimale Schulungsleistungen notwendig:

- Einweisung für Dozenten: Kurze Einführung (max. 1 Stunde), in der die Funktionsweise erläutert wird (Navigation, Lern-/Simulationsmodus, Interpretation der Ergebnisse).
- Selbsterklärende Nutzung für Lernende: Die Oberfläche ist so gestaltet, dass keine zusätzliche Schulung erforderlich ist. Hilfetexte und kontextsensitive Hinweise unterstützen die selbstständige Nutzung.

5.4 Web-API

Die Inhalte werden über eine REST-API im JSON-Format bereitgestellt.

Die Schnittstellen ermöglichen den Zugriff auf Kataloge und Fragen.

Geplante Endpunkte:

- `GET /api/kataloge` – liefert eine Liste aller Kataloge
- `GET /api/kataloge/{id}` – liefert alle Fragen eines bestimmten Katalogs
- `GET /api/fragen/{id}` – liefert Detailinformationen zu einer einzelnen Frage

Optionale Parameter:

- `?shuffle=true` – liefert Fragen oder Antwortoptionen in zufälliger Reihenfolge
- `?limit=10` – begrenzt die Anzahl der zurückgegebenen Fragen

Die API ist so aufgebaut, dass weitere Endpunkte (z. B. für neue Themenbereiche) einfach ergänzt werden können.

6. Abnahmekriterien

6.1 Qualitätsziele

Die Abnahme des Projekts erfolgt anhand klar definierter Qualitätsziele. Diese umfassen:

- Funktionale Vollständigkeit: Alle im Lastenheft aufgeführten Muss-Kriterien sind umgesetzt (zwei Modi, Fragetypen, Navigation, Feedback und Ergebnisübersicht).
- Benutzerfreundlichkeit: Die Oberfläche ist intuitiv bedienbar, responsiv und ermöglicht auch ungeübten Nutzer eine problemlose Nutzung. Nutzer sollen in der Lage sein, innerhalb weniger Minuten den Umgang mit der App selbstständig zu erlernen.
- Stabilität und Performance: Die Anwendung reagiert flüssig; Seiten- und Zustandswechsel erfolgen ohne merkbare Verzögerungen. Kataloge werden in der vorgesehenen Zeit (unter 5 Sekunden) geladen.
- Plattformunabhängigkeit: Die Anwendung ist auf allen gängigen Browsern und Endgeräten lauffähig und bietet konsistente Funktionalität.
- Inhaltliche Qualität: Fragen und Antworten sind korrekt, verständlich formuliert und vollständig. Das System interpretiert die Inhalte zuverlässig.
- Barrierefreiheit: Grundfunktionen sind barrierearm nutzbar (Tastaturbedienung, ausreichende Kontraste, klare Fokus-Indikatoren).

6.2 Testszenarien

Zur Abnahme werden konkrete Tests durchgeführt, die die Anforderungen abdecken. Beispiele:

- Test 1: Katalogauswahl – Ein Lernender kann ein Thema und einen Katalog auswählen; die Fragenliste wird korrekt angezeigt.

- Test 2: Lernmodus – Eine Frage wird angezeigt, die Lösung lässt sich kontrolliert aufdecken, Feedback und Erklärung erscheinen.
- Test 3: Simulation – Eine Prüfungssimulation mit Zeitlimit kann gestartet werden; während der Bearbeitung erscheint kein Feedback; am Ende wird eine vollständige Ergebnisübersicht angezeigt.
- Test 4: Fragetypen – Single-Choice, Multiple-Choice und Fill-in Fragen werden korrekt dargestellt und bewertet.
- Test 5: Navigation – Vor-/Zurücknavigation funktioniert zuverlässig; Rücksprung zur Übersicht ist möglich.
- Test 6: Performance – Laden eines kompletten Katalogs dauert unter 5 Sekunden bei einer 4G-Verbindung.
- Test 7: Plattformkompatibilität – Funktionstest auf aktuellen Versionen der gängigen Browser.
- Test 8: Fehlerfälle – Unterbrechung der Internetverbindung oder leere Kataloge führen zu klaren Fehlermeldungen mit Handlungsmöglichkeiten.
- Test 9: API-Schnittstellen
Ein GET-Aufruf an `/api/kataloge` liefert innerhalb von 2 Sekunden eine vollständige Liste der verfügbaren Kataloge.
Die Struktur entspricht dem definierten Datenmodell, sodass die App die Inhalte korrekt darstellen kann.