

Projektvision

Ziel ist die Entwicklung einer webbasierten Quiz-Applikation, die zwei Nutzer:innen in Echtzeit miteinander verbindet und durch interaktive Elemente wie Abstimmungen, Konsensfindung und Chat den Austausch fördert. Im Vordergrund steht ein leicht verständlicher und spielerischer Zugang, der Diskussion und gemeinsames Lernen unterstützt. Der Prototyp soll die Kernfunktionen eines kooperativen Quiz-Tools abbilden und dabei als Grundlage für mögliche spätere Erweiterungen dienen. Durch den Einsatz moderner Webtechnologien wird eine einfache Nutzung im Browser ohne Installation ermöglicht. Damit leistet das Projekt einen Beitrag zur Erprobung innovativer, digitaler Lern- und Kommunikationsformen.

Teamvorstellung



Kristin Leibling

B.Sc. Wirtschaftsinformatik, 5. Semester,

Master Data Management Specialist

Scrum Master, Termin- & Projektmanagement



Kathrin Kleine Büning

B.Sc. Wirtschaftsinformatik, 5. Semester,

Administrative Leitung im Consulting-Bereich

Projektleitung, Entwicklung, technische Dokumentation



Philipp Zybala

B.Sc. Wirtschaftsinformatik, 5. Semester,

Supply Specialist in der Energiebranche

Technische Leitung, Entwicklung, technische Dokumentation

Projektauftrag

Der Projektauftrag beschreibt die wesentlichen Rahmenbedingungen und Zielsetzungen für die Umsetzung des Online-Quizsystems. Er dient als Leitlinie für alle Projektbeteiligten und stellt sicher, dass Anforderungen, Lösungsansätze sowie organisatorische Aspekte klar definiert und nachvollziehbar dokumentiert sind. Damit bildet der Projektauftrag die Grundlage für die weitere Planung und Durchführung des Projekts.

Anforderungen

Um die gewünschte Funktionalität und Qualität des Quizsystems sicherzustellen, müssen die Anforderungen an das System ermittelt werden. Zu berücksichtigen sind die Interessen aller Stakeholder und die Systemumgebung. Quellen für Anforderungen sind der Projektauftrag, eigene Vorstellungen und Erfahrungen sowie Fremdsystem. Die Anforderungen gliedern sich in funktionale und nichtfunktionale Anforderungen, die die Basis für Konzeption, Entwicklung und spätere Tests bilden. Die hier aufgeführten Anforderungen sollen einen groben Rahmen abstecken und werden im Verlaufe des Projekts verfeinert.

Funktionale Anforderungen legen die konkreten Features und Abläufe des Systems fest:

- F1 Quiz mit Single-Choice-Fragen
- F2 Erstellung eines Quizraums für zwei Personen
- F3 Kooperativer Spielmodus
- F4 Feedback auf Antworten
- F5 Fragen können von allen Usern eingereicht, kommentiert und gemeldet werden
- F6 Registrierung/Login
- F7 Weitere Spielmodi (Wettkampf, >2 Spieler:innen).
- F8 Fragen können von allen Usern eingereicht, gemeldet und kommentiert werden

Nichtfunktionale Anforderungen betreffen vor allem Qualitätsmerkmale, technische Rahmenbedingungen und Benutzerfreundlichkeit:

- N1 Vollständig im Webbrowser nutzbar, keine Installation.
- N2 Responsiv für Desktop & Mobile.
- N3 Stabile Performance
- N4 Einfache, intuitive Bedienbarkeit
- N5 Zuverlässige Echtzeit-Synchronisation
- N6 Sichere Datenübertragung
- N7 Hosting auf kostenlosen Plattformen

Lösungsansatz

Ziel unseres Projekts ist die prototypische Umsetzung eines webbasierten Quizsystems, das Studierende der IU bei der Wiederholung und Vertiefung von Lerninhalten unterstützt.

Der Fokus des Prototyps liegt auf kooperativem Spielen mit zwei Teilnehmer:innen pro Raum sowie der Erstellung und Meldung von Fragen über eine separate Fragen-Seite. Für den Austausch zwischen den Spieler:innen soll eine Chatfunktion angeboten werden.

Die Anwendung soll direkt im Webbrowser nutzbar sein (ohne Installation) und den Self Service Gedanken unterstützen: Nutzer:innen können selbstständig Räume erstellen, Fragen einreichen und Fehler melden, ohne dass ein Administrator eingreifen muss.

Für die Erstellung der Webanwendung soll ein modularer Aufbau (React) gewählt werden, so dass spätere Erweiterungen jederzeit möglich sind. Das Backend wird durch Module wie Firebase oder Supabase unterstützt.

Grundstruktur des Systems

Das System besteht aus drei zentralen Komponenten:

Projekt Quiz – Projektkonfiguration (MS1)

- Frontend: Benutzeroberfläche mit React und Tailwind CSS
- Backend-as-a-Service: Datenspeicherung, Authentifizierung und Realtime-Synchronisation über Supabase (PostgreSQL)
- Hosting/Deployment: Bereitstellung der Web-App auf Vercel

Diese Architektur ermöglicht:

- Schnelle Entwicklung ohne eigene Server-Administration
- Modularen, wartbaren Code für zukünftige Erweiterungen
- Sofortige Nutzbarkeit im Browser

Hauptfunktionen des Prototyps

Spielraum & Kooperativer Modus:

- Erstellen eines Quizraums mit eindeutigem Raumcode
- Beitritt durch zweite Person per Raumcode (max. 2 Spieler:innen)
- Gleichzeitige Beantwortung von Single-Choice-Fragen
- Live-Anzeige der Antworten und Konsensfindung
- „Antwort zeigen“-Funktion mit richtiger Lösung und optionaler Begründung
- Timer pro Frage
- Punktevergabe für richtige Antworten
- Ergebnisübersicht am Ende der Runde

Fragenmanagement (separate Seite):

- Erstellung neuer Single Choice Fragen mit Pflichtfeldern (Frage, 2–5 Antwortoptionen, richtige Antwort, optionale Begründung)
- Validierung der Eingaben
- Fehler melden zu bestehenden Fragen (auf der Fragen Seite)
- Fehler melden zu gespielten Fragen direkt aus der Auswertung

Datenhaltung:

- Speicherung von Fragen, Meldungen, Spielergebnissen in Supabase

Kooperativer Aspekt

Der kooperative Aspekt zeigt sich auf zwei Ebenen:

1. Im Spiel:
 - a. Zwei Spieler:innen sehen live die Antworten des jeweils anderen.
 - b. Konsensanzeige unterstützt gemeinsames Entscheiden.

- c. Gemeinsame Punktwertung fördert Zusammenarbeit.
- d. Kommunikation über einen Live-Chat.
- 2. Bei der Inhaltsqualität:
 - a. Alle Nutzer:innen können neue Fragen erstellen.
 - b. Fehlerhinweise können gemeinsam gepflegt werden, um den Fragenpool langfristig zu verbessern.

Technologie-Stack

Ebene	Technologie
Frontend	React + Tailwind CSS
Backend / DB	Supabase (PostgreSQL, Auth, Realtime)
Hosting	Vercel
Projektmanagement	GitHub, Redmine (IU), Notion/ClickUp
Versionierung	GitHub
Video / Screencast	OBS

Erweiterbarkeit (Zukunftsaussichten)

Der modulare Aufbau erlaubt spätere Ergänzungen, wie z.B.:

- Weitere Spielmodi (Wettkampfmodus, Zeitduell, >2 Spieler:innen)
- Nutzerprofile mit Statistiken
- Teamfunktionen mit Einladungen
- KI-gestützte Fragevorschläge
- Kategoriefilter beim Quizstart

Abgrenzung für den Prototypen

Der erste Prototyp verzichtet bewusst auf:

- Gamification-Elemente (Badges, Level)
- Mehr als zwei Spieler:innen pro Raum
- Komplexe Teamfunktionen
- Asynchrones Spielen (nur Echtzeit)

Annahmen und Beschränkungen

Für die Planung und Umsetzung des Projekts ist es notwendig, bestimmte Annahmen zu treffen und Rahmenbedingungen festzulegen. Annahmen definieren die

Voraussetzungen, von denen im Projekt ausgegangen wird, während Beschränkungen die Grenzen und Einschränkungen des Prototyps aufzeigen. Gemeinsam bilden sie den Rahmen, innerhalb dessen die Projektziele realistisch erreicht werden können

Annahmen:

- A1 Die Nutzer:innen verwenden moderne Webbrowser (Chrome, Firefox, Edge, Safari) in aktueller Version.
- A2 Grundlegende Bedienkenntnisse von Webanwendungen werden vorausgesetzt.
- A3 Supabase im Free-Tier bietet ausreichende Performance und Speicherkapazität für den Prototyp (max. 2 gleichzeitige Nutzer:innen).
- A4 Es wird ausschließlich eine browserbasierte Web App entwickelt (keine native App).
- A5 Der Prototyp verwendet ausschließlich fiktive Beispiel Fragen (keine echten Prüfungsfragen oder urheberrechtlich geschützten Inhalte).
- A6 Ziel ist eine prototypische Demonstration der Kernfunktionalität, kein vollwertiges Produkt.
- A7 Die Organisation der Spielteilnehmer erfolgt ausschließlich über den generierten Raumcode (keine komplexen Teamsysteme im Prototyp).
- A8 Externe Dienste wie Vercel (Frontend Hosting) und Supabase (Backend, DB, Auth) werden in der Free-Tier-Variante genutzt.
- A9 Die Nutzer:innen interagieren zeitgleich im Raum; asynchrones Spielen ist nicht vorgesehen.
- A10 Erstellte Fragen und Meldungen werden direkt in der Supabase Datenbank gespeichert, ohne nachgelagerte manuelle Überprüfung.

Beschränkungen:

- B1 Kein Offline-Modus, durchgehender Internetzugang erforderlich.
- B2 Keine manuelle Prüfung oder Moderation von eingereichten Fragen im Prototyp (nur Speicherung von Meldungen).
- B3 Keine erweiterten Nutzerrollen (z. B. Admin oder Moderator).
- B4 App ist einsprachig (Deutsch).
- B5 Keine Integration mit IU-Systemen wie myCampus.
- B6 Keine DSGVO-Prüfung, da keine echten personenbezogenen Daten verarbeitet werden.
- B7 Prototyp umfasst nur den kooperativen Modus mit maximal 2 Spielern pro Raum.
- B8 Zusätzliche Funktionalitäten (z. B. Wettkampfmodus) werden nur als visuelle Teaser im Frontend dargestellt, ohne Backend-Logik.

- B9 Keine gleichzeitige Bearbeitung einer Frage durch mehrere Nutzer:innen (kein Echtzeit-Co-Editing).
- B10 Keine Unterstützung für mehr als einen aktiven Raum pro Nutzer:in im Prototyp.

Projektstrukturplan

Der Projektstrukturplan zeigt die wesentlichen Meilensteine, Arbeitspakete und deren geschätzten Zeitaufwände:

MS 1 – Projektkonfiguration abgeschlossen und bereitgestellt ~ 30 Std.

- 1.1 Projektvision erstellen: Formulierung der Zielsetzung, Nutzen und Motivation
- 1.2 Teamvorstellung aufbereiten: Namen, Fotos, Profilbeschreibungen, Skills zusammenstellen
- 1.3 Projektauftrag formulieren: Anforderungen, Lösungsansatz, Annahmen, Beschränkungen dokumentieren
- 1.4 Meilensteinplan erstellen: Zeitplan mit MS-Daten
- 1.5 Projektstrukturplan entwerfen: Gliederung in Arbeitspakete und Verantwortlichkeiten
- 1.6 Rollen & Verantwortungen definieren: Rollenbeschreibung, Verantwortlichkeiten, Zuständigkeiten
- 1.7 Technische Infrastruktur aufbauen: ClickUp/Teams/Versionsverwaltung einrichten
- 1.8 Ressourcenplanung & Urlaubskalender erstellen: Personalmanagement vorbereiten
- 1.9 Kommunikationsmanagement festlegen: Regelmeetings, Kanäle, Dokumentationsregeln
- 1.10 Risikomanagement initialisieren: Risiken identifizieren, Matrix anlegen

MS 2 – Projektvideo bereitgestellt ~ 10 Std

- 2.1 Storyboard & Skript: Inhaltliche Planung, Szenenabfolge
- 2.2 Aufnahmetechnik vorbereiten: Toolauswahl, Testaufnahme
- 2.3 Videoaufnahme: Vorstellung Projekt & Lösungsidee
- 2.4 Videoschnitt & Export: Bearbeitung, Untertitel, Export

MS 3 – Dokumentationskonzept bereitgestellt ~ 60 Std.

- 3.1 Struktur Dokumentationskonzept definieren: Kapitel, Inhalte, Format
- 3.2 Beschreibung Projektdokumente: Erstellung Backlogs, Risikoliste, Testprotokolle etc.
- 3.3 Implementierungsprozess dokumentieren: Vorgehensmodell, Rollenverteilung
- 3.4 Architektur und Modelle vorbereiten: EPK, UML, ERM-Skizzen
- 3.5 Qualitätsmanagementplan schreiben: Ziele, Tests, **Review-Verfahren**

MS 4 – Zentrale Ergebnisartefakte bereitgestellt ~ 120 Std.

- 4.1 Frontend entwickeln: UI, Navigation, Quizlogik Client
- 4.2 Backend entwickeln: API, Datenbank, Authentifizierung
- 4.3 Spielmodi implementieren: Kooperativ & kompetitiv
- 4.4 Content-Management-Funktionen: Fragen einpflegen, editieren, Versionierung
- 4.5 Integrationstests: Funktionale Tests, Bugfixing

MS 5 – Ergebnispräsentation als Screencast ~ 20 Std.

- 5.1 Präsentationskonzept: Ablauf, Inhalte, Storyline
- 5.2 Screencast aufnehmen: Demo, technische Erklärung
- 5.3 Schnitt & Finalisierung: Bearbeitung, Tonkorrektur

MS 6 – Finale Ergebnisse, Dokumentation & Projektbericht ~ 120 Std.

- 6.1 Enddokumentation fertigstellen: Benutzerhandbuch, Entwicklerdokumentation, Ergebnisbericht
- 6.2 Review & Qualitätscheck: Vollständigkeit, Formatierung, Konsistenz
- 6.3 Upload in Redmine / Turnitin: Endversion bereitstellen

Insgesamt wird für die Umsetzung des Projekts ein Gesamtaufwand von ca. 360 Stunden veranschlagt, was eine sorgfältige Planung und Priorisierung aller Aufgaben erforderlich macht. Die Fertigstellung des Projekts ist für den 26.01.2026 vorgesehen. Der zeitliche Ablauf ist im Gantt-Diagramm dargestellt, das im Anhang des Dokuments enthalten ist.

Rollen & Verantwortungen

Rolle	Verantwortlichkeiten	Name
Projektleiter	Gesamtkoordination des Projekts, Priorisierung von Aufgaben, Sicherstellung der Zielerreichung, Ansprechpartner für Dozenten/Betreuer.	Kathrin Kleine Büning
Zeit- und Projektplanungs-verantwortlicher	Erstellung und Pflege des Projektzeitplans, Überwachung von Deadlines, Terminplanung, frühzeitiges Erkennen und Kommunizieren von Verzögerungen, Abstimmung mit allen Rollen zur Einhaltung der Meilensteine.	Kristin Leibling
Technischer Leiter / Softwarearchitekt	Definition der Softwarearchitektur, Auswahl von Technologien und Frameworks, Sicherstellung der Codequalität, Unterstützung bei technischen Entscheidungen.	Philipp Zybala
Entwickler	Umsetzung des Benutzerinterfaces (UI) im Browser, Implementierung von	Kathrin Kleine

	Designvorlagen, Sicherstellung der Responsiveness und Usability Entwicklung der Serverlogik, APIs, Datenbankbindung und Authentifizierung, Implementierung der Geschäftslogik.	Büning, Philipp Zybala
UI/UX-Designer	Gestaltung des User Interfaces, Erstellung von Mockups/Wireframes, Sicherstellung der Nutzerfreundlichkeit und Designkonsistenz.	Kathrin Kleine Büning
Requirements Engineer	Erfassung und Dokumentation der Anforderungen, Sicherstellung von Klarheit und Konsistenz.	Philipp Zybala
QA-Tester / Qualitätssicherung	Erstellung von Testplänen, Durchführung von Funktionstests, Dokumentation und Meldung von Fehlern, Retests nach Bugfixes.	Alle
Dokumentations-verantwortlicher	Erstellung und Pflege der Projektdokumentation, Vorbereitung von Präsentationsunterlagen.	Kristin Leibling
Kommunikationsmanager	Verwaltung der Kommunikationskanäle (Teams, WhatsApp, ClickUp), Organisation der Meetingstruktur, Pflege der Informationsdokumentation.	Kristin Leibling
Risikomanager	Identifikation möglicher Projektrisiken, Entwicklung von Maßnahmenplänen, Monitoring von kritischen Projektbereichen.	Kathrin Kleine Büning

Aufbau der technischen Infrastruktur

Die technische Infrastruktur des Projekts basiert auf der Kombination moderner Cloud-Technologien und Tools, die eine schnelle Entwicklung, einfache Bereitstellung und Zusammenarbeit im Team ermöglichen. Im Folgenden werden die einzelnen Komponenten und ihre Rolle im Projekt erläutert.

App

Die Anwendung besteht aus drei zentralen technischen Komponenten:

- **Frontend:** Die Benutzeroberfläche wird mit React entwickelt und durch Tailwind CSS für ein modernes, responsives Design ergänzt. Sie bildet die Grundlage für alle Nutzerinteraktionen. Vom Erstellen von Fragen bis zum Spielen im kooperativen Modus.
- **Backend-as-a-Service:** Anstelle eines selbst entwickelten Backends wird Supabase eingesetzt. Supabase übernimmt die Speicherung von Daten, die

Authentifizierung (Login) sowie Echtzeit-Funktionen, um Antworten und Spielstände zwischen zwei Teilnehmer:innen synchron zu halten.

- Datenbank: Supabase stellt eine PostgreSQL-Datenbank bereit. Darin werden u. a. Fragen, Spiele, Antworten, Meldungen und optional Nutzerdaten verwaltet. Die Datenbank ist cloudbasiert und benötigt keine lokale Installation.

Verwendete Software

Zur Unterstützung der Entwicklung und Verwaltung des Projekts kommen folgende Tools und Plattformen zum Einsatz:

- Visual Studio Code: Hauptentwicklungsumgebung für die Programmierung der React Komponenten und die Integration von Supabase. Die IDE wird durch Erweiterungen für JavaScript, Tailwind CSS und Git ergänzt.
- Vercel: Hosting-Plattform für das Frontend. Jeder Push ins GitHub-Repository löst automatisch einen Build-Prozess aus und veröffentlicht die aktuelle Version der App.
- Supabase Web Console: Grafische Benutzeroberfläche zum Verwalten der Datenbank, Authentifizierungseinstellungen und Realtime-Funktionen. Hier können Tabellen, Policies und Backups konfiguriert werden.
- GitHub: Versionsverwaltung und zentrale Codebasis des Projekts. Ermöglicht kollaboratives Arbeiten, Code-Reviews und eine nahtlose Anbindung an Vercel.
- Office365: Erstellung und Bearbeitung von Projektdokumentation (Word) sowie Präsentationen (PowerPoint) für Zwischenergebnisse und die Endpräsentation.
- Redmine (IU): Projektplanung, Dokumentenablage und Meilensteinverfolgung entsprechend den IU-Vorgaben
- ClickUp: Scrum Verwaltung

Personalmanagement

Ressourcenplanung

Die verfügbaren Kapazitäten der Teammitglieder werden anhand des Projektplans auf die Arbeitspakete verteilt. Dabei wird ein Gesamtaufwand von ca. 360 Stunden berücksichtigt, der gleichmäßig und unter Berücksichtigung von Abwesenheiten auf das Team verteilt wird. Rollen und Verantwortlichkeiten stellen sicher, dass alle zentralen Aufgabenbereiche abgedeckt sind.

Urlaub / Abwesenheitsplanung

Abwesenheiten werden mindestens 1 Woche vorher in der Teams-Gruppe gemeldet. Vertretung wird individuell abgesprochen und im Abwesenheitsplan eingetragen.

Name	Geplante Abwesenheit	Zeitraum
Philipp Zybala	Urlaub	01.-14.09.2025
Alle	Weihnachten	24.12.-31.12.2025

Kommunikationsmanagement

Zur Steuerung des Projekts wird ein hybrider Ansatz aus Microsoft Teams, WhatsApp, ClickUp und Scrum-Meetings genutzt. Die zentrale Abstimmung erfolgt in wöchentlichen Scrum-Meetings (Sprint Planning, Reviews und Retrospektiven), die Montags von 20:00 bis 21:30 Uhr über Microsoft Teams stattfinden. Ziel ist es, den Fortschritt transparent zu machen, Hindernisse zu besprechen und die nächsten Schritte im Team zu planen.

Für den laufenden Austausch steht Microsoft Teams als Hauptplattform zur Verfügung – sowohl für Besprechungen als auch für die gemeinsame Dateiverwaltung und Protokollierung von Entscheidungen. Ergänzend wird eine WhatsApp-Gruppe für schnelle Abstimmungen und spontane Rückfragen genutzt, wobei wichtige Informationen zusätzlich in Teams dokumentiert werden, um eine langfristige Nachvollziehbarkeit sicherzustellen.

Zur Aufgabenorganisation wird ClickUp eingesetzt: Das Kanban-Board visualisiert die aktuellen User Stories und deren Bearbeitungsstatus (Backlog, In Bearbeitung, Erledigt), sodass alle Teammitglieder jederzeit den Fortschritt einsehen können. Ergänzend dient das Gantt-Diagramm zur Darstellung der zeitlichen Planung und Abhängigkeiten der Arbeitspakete, um Engpässe oder Verzögerungen frühzeitig zu erkennen und gezielt gegenzusteuern.

Risikomanagement

Zur Sicherstellung eines erfolgreichen Projektverlaufs wurde eine detaillierte Risikoliste erstellt. Diese listet potenzielle Risiken in verschiedenen Kategorien wie Ressourcen-, Termin-, Akzeptanz- und Qualitätsrisiken auf.

Für jedes identifizierte Risiko sind die möglichen Auswirkungen, geeignete Risikostrategien sowie die jeweilige Zuständigkeit dokumentiert. Dadurch wird sichergestellt, dass Risiken frühzeitig erkannt und geeignete Maßnahmen zur Vermeidung oder Verminderung eingeleitet werden können.

Die Tabelle dient als zentrales Werkzeug im Projektcontrolling: Sie wird regelmäßig überprüft und bei Bedarf ergänzt, um auf neue Erkenntnisse oder Veränderungen im Projektverlauf reagieren zu können. Durch diesen strukturierten Ansatz wird das Risiko von Terminverzögerungen, Qualitätsverlusten oder unzureichender Zielerreichung deutlich reduziert.

Anhang

Risikomanagement.xls

GANTT Diagramm Redmine