



Dokumentation zur betrieblichen Projektarbeit

Fachinformatiker Anwendungsentwicklung

Abschlussprüfung Sommer 2025

Entwicklung eines Prototyps
zur Durchführung medizinischer
Umfragen mit dynamischer Fragenlogik

Autor

Justin Lewczuk

Prüflingsnummer: 0000701667

E-Mail: Justin-Lewczuk@hotmail.de

Bearbeitungszeitraum: 05.05.2025 – 16.05.2025

Abgabetermin 26.05.2025

Ausbildungsbetrieb

GFN GmbH
Hammerbrookstraße 90
20097 Hamburg

Ausbilder im Betrieb

Martin Krause
hamburg@gfn.de

Praktikumsbetrieb

TICOH GmbH
Klinikweg 23
22081 Hamburg

Projektbetreuer

Markus Habetha
markus.habetha@ticoh.de

Inhaltsverzeichnis

| | |
|--|----|
| Abbildungsverzeichnis | 2 |
| Tabellenverzeichnis | 2 |
| Glossar | 3 |
| 1. Einleitung | 4 |
| 1.1 Projektbeschreibung | 4 |
| 1.2 Projektziel | 4 |
| 1.3 Projektschnittstellen | 5 |
| 1.4 Projektbegründung..... | 5 |
| 1.5 Projektabgrenzung..... | 6 |
| 2. Projektplanung | 6 |
| 2.1 Projektphasen..... | 6 |
| 2.2 Ressourcenplanung | 7 |
| 2.3 Entwicklungsprozess | 7 |
| 3. Analysephase | 8 |
| 3.1 IST-Analyse | 8 |
| 3.2 Nutzwertanalyse | 9 |
| 3.3 Marktanalyse | 9 |
| 3.4 Entscheidungsgrundlage | 10 |
| 3.5 Lastenheft..... | 10 |
| 4. Entwurfsphase..... | 10 |
| 4.1 Architekturdesign | 10 |
| 4.2 Entwurf und Gestaltung der Benutzeroberfläche | 11 |
| 4.3 Datenmodell..... | 12 |
| 4.4 Maßnahmen zu Qualitätssicherung..... | 13 |
| 4.5 Pflichtenheft | 14 |
| 5. Implementierungsphase | 14 |
| 5.1 Implementierung der Datenstrukturen | 15 |
| 5.2 Dynamische Umfrageoberfläche mit Anzeige-Logik | 16 |
| 5.3 Backend Logik und Schnittstellen | 17 |
| 6. Projektabschluss | 18 |
| 6.1 Zusammenfassung der Umsetzung | 18 |
| 6.2 Test und Qualitätssicherung | 18 |
| 6.3 Lessons Learned | 18 |
| 6.4 Ausblick..... | 19 |
| 7. Danksagung..... | 19 |

| | |
|--|---|
| Anhang | 20 |
| A. Detaillierte Zeitplanung der Projektphasen..... | 20 |
| B. Beispiel Use Case Diagramm..... | 20 |
| C. Marktanalyse und Entscheidungsmatrix | 21 |
| D. Auszug Lastenheft | 22 |
| E. Auszug Pflichtenheft | 22 |
| F. Beispielhafte GUI-Ansicht des Prototyps | 23 |
| G. Projektabnahme | 24 |
| H. Programmcode Prototyp Umfragetool..... | Fehler! Textmarke nicht definiert. |

Abbildungsverzeichnis

| | |
|--|----|
| Abbildung 1.: Struktur der Datenbanktabelle im Umfragetool (PHPMyAdmin). | 15 |
| Abbildung 2.: Teilnehmerantworten, Gruppenzuordnung und Zeitstempel (PHPMyAdmin). | 15 |
| Abbildung 3.: Konfiguration des Express-Servers (VS Code). | 16 |
| Abbildung 4.: Backend-Route zur Bereitstellung der Umfragefragen (VS Code). | 16 |
| Abbildung 5.: Verarbeitung der übermittelten Antworten zur Speicherung in Datenbank.(VS Code). | 16 |
| Abbildung 6.: Einbindung der modularen API-Routen im Express-Backend (VS Code). | 17 |
| Abbildung 7.: Serverseitige Verarbeitung und Speicherung der Umfrageantworten (VS Code). | 17 |
| Abbildung 8.: Überprüfen des Tokens auf Gültigkeit und Nutzung vor Umfragestart (VS Code). | 17 |
| Abbildung 9.: Detaillierte Zeitplanung Projektphasen (Excel) | 20 |
| Abbildung 10.: Anwendungsfalldiagramm - Umfragetool (draw.io) | 20 |
| Abbildung 11.: Marktanalyse 1 | 21 |
| Abbildung 12.: Marktanalyse 2 | 21 |
| Abbildung 13.: Bewertungsmatrix (Excel) | 21 |
| Abbildung 14.: Darstellung einer Umfragefrage mit Token basiertem Zugriff (Prototypansicht). | 23 |
| Abbildung 15.: Übersicht einer Umfrageeinreichung mit zugehörigem Token im Adminbereich | 23 |

Tabellenverzeichnis

| | |
|--|----|
| Tabelle 1.: Projektphasen Zeitplanung | 7 |
| Tabelle 2.: Ressourcenplanung | 7 |
| Tabelle 3.: Vergleich Question Pro vs. Umfragetool | 9 |
| Tabelle 4.: Endpunkte Server API | 11 |
| Tabelle 5.: fragen Tabelle MySQL | 12 |
| Tabelle 6.: antworten Tabelle MySQL | 13 |
| Tabelle 7.: fragebogen_status Tabelle MySQL | 13 |

Glossar

| Begriff | Erklärung |
|------------|--|
| KIM | Kommunikation im Medizinwesen -ein sicherer E-Mail-Dienst in der Telematikinfrastruktur |
| VSDM | Versichertenstammdatenmanagement - Online Abgleich der Patientendaten |
| eRezept | Das Elektronische Rezept -neu im Gesundheitswesen |
| Token | Ein einmalig nutzbarer Zugangscode. |
| UUID | Universal Unique Identifier -ein eindeutiger Bezeichner. |
| API | Application Programmin Interface -eine Programmierschnittstelle die den Austausch zwischen Systemen ermöglicht |
| REST-API | Eine API die auf den Prinzipien von HTTP basiert. |
| Frontend | Der sichtbare Teil der Anwendung im Browser, den Benutzer bedienen. |
| Backend | Der serverseitige Teil, der Anfragen verarbeitet werden und Daten speichert. |
| JSON | Datenformat zur strukturierten Übertragung von Informationen zwischen Frontend und Backend |
| POST-Route | Eine HTTP-Methode, mit der Daten an den Server gesendet werden. |
| MySQL | Ein relationales Datenbanksystem zur strukturierten Speicherung von Daten |
| HTTP | Ein Protokoll, das die Grundlage für die Kommunikation zwischen Webbrowser und Webserver bildet. |

1. Einleitung

Diese Projektdokumentation beschreibt die Konzeption und prototypische Umsetzung eines webbasierten Umfragetools im Rahmen des Abschlussprojekts zum Fachinformatiker für Anwendungsentwicklung bei der TICOH GmbH. Das Projekt wurde vom Prüfling eigenständig initiiert und in Abstimmung mit dem Praktikumsbetrieb durchgeführt.

Die TICOH GmbH führt regelmäßig Umfragen unter medizinischen Einrichtungen durch, um beispielsweise den Stand bei der Einführung von Telematikinfrastruktur-Diensten (z.B. Kim, eRezept, VSDM) zu erfassen. Die bislang eingesetzten externen Umfragetools boten nur eingeschränkte Anpassungs- und Integrationsmöglichkeiten. Vor diesem Hintergrund entstand die Idee, einen eigenen Umfragetool-Prototypen zu entwickeln, um zentrale Anforderungen zu validieren und eine Entscheidungsgrundlage für eine spätere Make-or-Buy-Strategie zu schaffen.

1.1 Projektbeschreibung

Im Rahmen des Praktikums bei der TICOH GmbH wurde die eigenverantwortliche Entwicklung eines webbasierten Umfragetool-Prototyps umgesetzt. Ziel war es, zentrale technische und funktionale Anforderungen zu prüfen, die in bisherigen Tools nur unzureichend erfüllt wurden. Dazu zählen unter anderem eine anpassbare Fragenlogik, die DSGVO-konforme Verarbeitung personenbezogener Daten sowie die Möglichkeit einer späteren Anbindung an ein CRM-System.

Die entwickelte Anwendung ermöglicht es, Umfragen über Zugriffslinks aufzurufen, Fragen dynamisch anzuzeigen und Ergebnisse strukturiert in einer Datenbank zu speichern.

Technologisch basiert das Projekt auf einem modularen Webstack bestehend aus einem Express.js-Backend mit REST-API, einer MySQL-Datenbank und einem browserbasierten Frontend, das die Umfrage im JSON-Format dynamisch darstellt. Die Anwendung ist für die Nutzung auf verschiedenen Endgeräten optimiert und ermöglicht den Zugriff auf Umfragen über individuelle Zugangslinks, ohne dass eine Registrierung erforderlich ist.

1.2 Projektziel

Ziel des Projekts war die prototypische Umsetzung zentraler technischer Anforderungen der TICOH GmbH, um auf dieser Basis eine fundierte Make-or-Buy-Entscheidung vorbereiten zu können. Der entwickelte Prototyp sollte konkret aufzeigen, inwiefern sich spezifische Anforderungen technisch umsetzen lassen und welche Erkenntnisse sich daraus für eine spätere Auswahl oder Weiterentwicklung eines geeigneten Umfragetools ableiten lassen.

Die spezifischen Ziele des Projekts umfassten:

1. **Benutzerfreundlichkeit:** Entwicklung einer intuitiven Benutzeroberfläche, die es ermöglicht, Umfragen einfach zu erstellen und zu bearbeiten.
2. **Dynamische Fragenlogik:** Implementierung von Funktionen, die es erlauben, Folgefragen basierend auf vorherigen Antworten anzuzeigen oder auszublenden.
3. **Datenschutzkonformität:** Sicherstellung, dass alle gesammelten Daten gemäß den Anforderungen der DSGVO gespeichert und verarbeitet werden.
4. **Integration in CRM-System:** Vorbereitung des Umfragetools für eine zukünftige Integration in ein CRM-System, um eine zentrale Verwaltung und Auswertung der Umfragedaten zu ermöglichen.
5. **Skalierbarkeit:** Gestaltung der Softwarearchitektur so, dass sie leicht erweitert und an zukünftige Anforderungen angepasst werden kann.

1.3 Projektschnittstellen

Im Verlauf der Projektumsetzung ergaben sich mehrere organisatorische Schnittstellen innerhalb des Unternehmens sowie zu externen Partnern. Diese Schnittstellen betrafen sowohl die Planung, Abstimmung als auch die spätere Nutzung des entwickelten Umfragetools.

Interne Schnittstellen

1. **Projektleitung(Prüfling)**
Ich habe das Projekt in der Rolle der Projektleitung eigenständig durchgeführt. Dabei habe ich die Anforderungen definiert, technische Entscheidungen getroffen und die Umsetzung selbstständig geplant und dokumentiert. Mein Ausbilder stand mir beratend zur Seite, die fachliche Verantwortung lag jedoch bei mir.
2. **Marketingabteilung**
Die Marketingabteilung plant, das Tool zur Durchführung von Umfragen bei Kunden in medizinischen Einrichtungen einzusetzen. Hier liegt eine zentrale organisatorische Schnittstelle in der Nutzung und Auswertung der Ergebnisse für strategische Zwecke.
3. **Personalabteilung**
Intern besteht weiteres Potenzial zur Nutzung für Mitarbeiterbefragungen. Die Personalabteilung wurde als möglicher zukünftiger Anwenderkreis in die Konzeption einbezogen.
4. **IT-Abteilung/Technische Ansprechpartner**
Für Fragen zur Integration in die bestehende IT-Infrastruktur (z. B. spätere CRM-Anbindung) gab es Abstimmungen mit technischen Ansprechpartnern innerhalb der TICOH GmbH.

Externe Schnittstellen

1. **Medizinische Einrichtungen / Kunden**
Die Zielgruppe des Tools sind Einrichtungen des Gesundheitswesens in Modellregionen der TI-Einführung. Diese externen Stakeholder stehen im Fokus der Umfragen und sind damit direkt in den späteren Einsatz des Tools eingebunden.
2. **Projektpartner/Modellregion**
Im Rahmen laufender Projekte zur Digitalisierung im Gesundheitswesen arbeitet die TICOH GmbH mit verschiedenen Akteuren (z. B. Kassenärztliche Vereinigungen, gematik, Systemhersteller) zusammen. Die gewonnenen Umfrageergebnisse können hier ebenfalls als Entscheidungsgrundlage genutzt werden.

1.4 Projektbegründung

Die Entscheidung zur Entwicklung eines Prototyps für ein Umfragetool für die TICOH GmbH basierte auf mehreren betrieblichen Anforderungen und strategischen Überlegungen:

1. **Make-or-Buy-Entscheidung:** Im Vorfeld wurde geprüft, ob ein bestehendes Umfragetool verwendet oder selbst entwickelt werden sollte. Da externe Tools nicht alle Anforderungen der TICOH GmbH erfüllten, wurde entschieden, zunächst einen eigenen Prototypen zu realisieren. Dieser dient als Grundlage für eine spätere strategische Entscheidung über eine langfristige Eigenentwicklung oder den Einsatz externen Lösungen.
2. **Datenschutz und Compliance:** Die Verarbeitung sensibler Daten erfordert strikte Einhaltung der Datenschutzgrundverordnung (DSGVO). Durch die Eigenentwicklung kann die TICOH GmbH sicherstellen, dass alle datenschutzrechtlichen Anforderungen erfüllt werden und die Kontrolle über die Datenverarbeitung im eigenen Haus bleibt.
3. **Integration in ein zukünftiges CRM-System:** Ein zentrales Ziel ist die nahtlose Integration des Umfragetools in ein geplantes Customer-Relationship-Management-System. Dies ermöglicht eine zentrale Verwaltung von Kontaktdaten, Umfrageergebnissen und Kommunikationshistorien, wodurch Prozesse effizienter gestaltet und Synergien genutzt werden können.

4. **Benutzerfreundlichkeit und Barrierefreiheit:** Das neue Tool soll eine intuitive Benutzeroberfläche bieten, die es medizinischen Einrichtungen ermöglicht, ohne technische Vorkenntnisse an Umfragen teilzunehmen. Dies erhöht die Teilnahmequote und verbessert die Qualität der gesammelten Daten.
5. **Skalierbarkeit und Zukunftssicherheit:** Die Eigenentwicklung ermöglicht es, das Tool modular aufzubauen und bei Bedarf, um zusätzliche Funktionen zu erweitern. So kann die TICOH GmbH flexibel auf zukünftige Anforderungen reagieren und das Tool kontinuierlich weiterentwickeln.

Durch den entwickelten Prototyp steht der TICO GmbH ein Testsystem zur Verfügung, mit dem sich zentrale Anforderungen an ein Umfragetool evaluieren lassen. Die daraus abgeleiteten Erkenntnisse unterstützen die Entscheidung, ob ein externes Tool angeschafft oder eine interne Lösung weiterentwickelt werden soll -inklusive der möglichen Anbindung an ein künftiges CRM-System.

1.5 Projektabgrenzung

Im Rahmen dieses Projekts lag der Fokus auf der Entwicklung eines webbasierten Umfragetools, das die Erstellung und Durchführung von Umfragen ermöglicht. Dabei wurden folgende Aspekte bewusst ausgeklammert, um den Projektumfang klar zu definieren und die Ressourcen effizient zu nutzen:

1. **Benutzerverwaltung:** Es wurde keine umfassende Benutzerverwaltung implementiert. Die Teilnahme an Umfragen erfolgt anonym über individuelle Zugangslinks, ohne die Notwendigkeit einer Benutzerregistrierung oder -authentifizierung.
2. **Erweiterte Sicherheitsmaßnahmen:** Das Projekt konzentrierte sich auf grundlegende Sicherheitsaspekte wie die Validierung von Eingaben und den Schutz vor unbefugtem Zugriff. Erweiterte Sicherheitsfunktionen wie Zwei-Faktor-Authentifizierung oder Verschlüsselung sensibler Daten wurden nicht implementiert.
3. **Design und Benutzeroberfläche:** Die Gestaltung der Benutzeroberfläche wurde funktional gehalten, mit dem Ziel, eine einfache und intuitive Bedienung zu gewährleisten. Auf aufwendige Designaspekte oder die Optimierung für verschiedene Endgeräte wurde verzichtet.
4. **Auswertung und Berichterstattung:** Die automatisierte Auswertung von Umfrageergebnissen und die Generierung von Berichten waren nicht Bestandteil dieses Projekts. Die gesammelten Daten werden in der Datenbank gespeichert und können bei Bedarf manuell ausgewertet werden.
5. **Integration in bestehende Systeme:** Obwohl das Umfragetool so konzipiert wurde, dass eine zukünftige Integration in ein CRM-System möglich ist, wurde diese Integration im Rahmen dieses Projekts nicht umgesetzt.

Diese Abgrenzungen ermöglichten es, sich auf die Kernfunktionalitäten des Umfragetools zu konzentrieren und eine solide Grundlage für zukünftige Erweiterungen zu schaffen.

2. Projektplanung

Das weitere Vorgehen wurde durch Planung der Projektphasen, Ermitteln der benötigten Ressourcen sowie Festlegen des Entwicklungsprozesses bestimmt.

2.1 Projektphasen

Zur Umsetzung des Projekts wurden 80 Stunden angesetzt, die in die verschiedenen Projektphasen aufgeteilt wurden. Eine grobe Zeitplanung der verschiedenen Projektphasen kann der folgenden Tabelle entnommen werden. Eine detaillierte Zeitplanung aller Projektphasen und deren Aufgabe kann dem Anhang A: Detaillierte Zeitplanung entnommen werden.

| Phasen | Zeit in Stunden |
|-----------------|-----------------|
| Analyse | 10 |
| Entwurf | 10 |
| Implementierung | 40 |
| Dokumentation | 20 |
| Summe | 80 |

Tabelle 1.: Projektphasen Zeitplanung

2.2 Ressourcenplanung

Alle zum Umsetzen der Anforderung erforderlichen Ressourcen wurden in der folgenden Tabelle aufgeführt. Neben Hard- und Software Ressourcen wurde auch das Personal mit aufgenommen. Im Hinblick auf anfallende Kosten wurde darauf geachtet, dass die Nutzung der Software kostenfrei ist oder die Lizenzen dem Unternehmen bereits zur Verfügung stehen. Dadurch konnten die Projektkosten auf ein Minimum gehalten werden.

| Ressource | | Rolle |
|-----------|---------------------|--|
| Personal | | |
| | Auszubildender | Projektdurchführung/ Dokumentation |
| | Fachlicher Betreuer | Code-Review/ technische Rücksprache |
| | Projektleiter | Abstimmung Feedback zur fachlichen Funktion |
| Hardware | | |
| | Notebook | Arbeits/ Testgerät |
| Software | | |
| | Windows 11 | Betriebssystem |
| | Visual Studio Code | Entwicklungsumgebung |
| | Microsoft Edge | Browser für Test und Bedienung der Webanwendung |
| | MySQL Workbench | Verwaltung der MySQL- Datenbank |
| | Postman | Testtool für API-Endpunkte |
| | Node.js + Express | JavaScript-Laufzeit und Web- Framework |
| | Draw.io | Erstellung ER-Modell, Prozessgrafiken |

Tabelle 2.: Ressourcenplanung

2.3 Entwicklungsprozess

Zur Umsetzung des Projekts wurde das klassische Wasserfallmodell verwendet, wie im Projektantrag definiert. Dabei wurde das Vorhaben in aufeinander aufbauende Phasen unterteilt: Analyse, Entwurf, Implementierung, Test und Dokumentation. Dieses strukturierte Vorgehen ermöglichte eine klare Planung und Umsetzung der einzelnen Arbeitsschritte innerhalb der vorgegebenen Projektzeit.

Obwohl das Wasserfallmodell als methodischer Rahmen diente, wurde in der Praxis mit regelmäßigen Feedbackrunden gearbeitet. In enger Abstimmung mit dem fachlichen Betreuer der TICOH GmbH konnten technische Details wie die Fragenlogik, die Datenbankstruktur oder die Steuerung der Antwortbarkeit auch innerhalb einer Phase angepasst oder erweitert werden. Diese Rücksprachen fanden überwiegend mündlich statt und ermöglichten eine praxisnahe, zielgerichtete Umsetzung.

Die Umsetzung erfolgte iterativ in kleinen Funktionsbausteinen: Zuerst wurden die grundlegenden API-Funktionen (z. B. GET /api/fragen), dann die Datenbankspeicherung und schließlich die Antwortverarbeitung entwickelt und getestet.

Ziel des gewählten Entwicklungsprozesses war es, ein stabiles und modular aufgebautes System zu schaffen, das einerseits der Planung nach dem Wasserfallmodell entspricht und andererseits flexibel auf betriebliche Anforderungen reagieren kann.

3. Analysephase

In der Analysephase meines Projekts zur Entwicklung eines prototypischen Umfragetools für die TICOH GmbH lag der Fokus darauf, die fachlichen und technischen Anforderungen möglichst genau zu erfassen. Ein zentraler Bestandteil war der regelmäßige Austausch mit meinem fachlichen Ansprechpartner im Praktikumsbetrieb, um die Zielsetzung der Anwendung, die funktionalen Anforderungen und mögliche spätere Erweiterungen (z.B. Integration in ein CRM-System) gemeinsam zu erarbeiten.

Die Anforderungen wurden auf Basis konkreter Anwendungsszenarien entwickelt: TICOH möchte regelmäßig Einrichtungen des Gesundheitswesens zu ihrem Stand im Bereich der Telematikinfrastruktur befragen beispielsweise zur Nutzung von KIM, eRezept oder ePA. Die bisher eingesetzten externen Tools boten jedoch kaum Möglichkeiten zur Individualisierung der Umfragen oder zur Erweiterung der Datenverarbeitung.

Im Rahmen dieser Phase wurde eine IST-Analyse durchgeführt, in der die Schwächen der bisherigen Lösung identifiziert wurden (z.B. keine Steuerung von Folgefragen, unflexible Auswertung, externe Datenhaltung). Aufbauend darauf wurde eine Nutzwertanalyse erstellt, die den Mehrwert eines eigens entwickelten Systems darstellte, insbesondere im Hinblick auf Datenschutz, Erweiterbarkeit und spätere Systemintegration.

Darüber hinaus wurden Beispiel-Use-Cases definiert, um typische Nutzerinteraktionen zu beschreiben etwa das Ausfüllen einer Umfrage durch eine medizinische Einrichtung oder die Verarbeitung der Antworten im Backend. Diese Anwendungsfälle bildeten die Grundlage für die weitere Planung von Datenstruktur, Fragenlogik und API-Routen. *Ein exemplarisches Use Case ist im Anhang dargestellt. (Siehe Anhang B)*

Abschließend wurde ein Lastenheft erstellt, das alle wesentlichen funktionalen und nicht-funktionalen Anforderungen zusammenfasst und als Grundlage für das Pflichtenheft sowie die technische Umsetzung diente.

3.1 IST-Analyse

In der aktuellen Situation nutzt die TICOH GmbH das externe Umfragetool QuestionPro zur Durchführung von Umfragen, beispielsweise zur Erhebung von Informationen zur Telematikinfrastruktur in medizinischen Einrichtungen. Die Umfragen werden dort erstellt, verteilt und ausgewertet. Zwar bietet QuestionPro grundlegende Funktionen wie Fragetypen, Bedingungen und Verzweigungslogik, jedoch sind diese nur über die grafische Oberfläche konfigurierbar und lassen sich nicht programmatisch oder strukturiert über ein zentrales Datenmodell steuern.

Ein wesentliches Problem liegt darin, dass die Anzeige- und Steuerungslogik der Fragen nicht im JSON-Format definiert oder über eine API verwaltet werden kann was jedoch ein zentrales Ziel des geplanten Umfragetools ist. Dadurch ist es nicht möglich, die Umfragen vollständig dynamisch und kontrolliert über das eigene System zu steuern oder in andere Systeme (z.B. ein CRM) zu integrieren.

Die aktuelle Lösung führt zu weiteren Herausforderungen:

- Medienbrüche und manuelle Zwischenschritte (z. B. beim Export oder der Auswertung)
- begrenzte Anpassbarkeit bei der Darstellung und Reihenfolge der Fragen
- fehlende Möglichkeit zur Erweiterung oder Automatisierung der Datenverarbeitung

Diese IST-Situation verdeutlicht den Bedarf nach einem internen, modularen Umfragetool, das die Daten strukturiert erfasst, vollständig in eigener Kontrolle verarbeitet und mittelfristig in ein CRM-System integrierbar ist. Ziel ist eine zentrale, sichere und technisch erweiterbare Lösung, die den spezifischen Anforderungen der TICOH GmbH entspricht.

3.2 Nutzwertanalyse

Auf Basis der IST-Analyse wurde eine Nutzwertanalyse durchgeführt, um die Vorteile einer zukünftigen Lösung im Vergleich zur bisherigen Nutzung von externen Tools wie QuestionPro herauszuarbeiten. Dabei wurden sowohl funktionale als auch nicht-funktionale Anforderungen berücksichtigt, die für den Einsatz im Umfeld der TICOH GmbH besonders relevant sind.

Die folgende Tabelle zeigt einen Vergleich zentraler Kriterien zwischen der bisherigen Lösung (QuestionPro) und dem geplanten Umfragetool:

| Kriterium | Bisheriges Tool | Geplantes Umfragetool |
|---------------------------------|---|---|
| Fragenlogik Bedingungen | Eingeschränkt über GUI, keine strukturierte Steuerung | Steuerbar über dynamische Logik im Frontend |
| Anpassbarkeit / Erweiterbarkeit | Nur im Rahmen der Plattformfunktionen | Modularer Aufbau, jederzeit erweiterbar |
| API-Integration / CRM-Anbindung | Nur begrenzt möglich / nicht vorgesehen | Von Anfang an vorbereitet für spätere CRM-Integration |
| Kosten / Lizenzmodell | Hohe Kosten, beim derzeitigen Anbieter | geringere Lizenzkosten, mehr Eigenkontrolle |
| Benutzerführung (Zugriffslogik) | Kein internes Kontrollsystem für Einmal-Links | Integrierte Logik zur Steuerung von Zugriffsrechten |
| Technologische Kontrolle | Eingeschränkter Zugriff auf Code und Datenmodelle | Volle Kontrolle über Backend, Datenbank und Frontend |

Tabelle 3.: Vergleich Question Pro vs. Umfragetool

Bewertung und Fazit

Die Analyse zeigt, dass das geplante Umfragetool in nahezu allen relevanten Bereichen einen höheren Nutzwert bietet. Besonders hervorzuheben ist die vollständige Kontrolle über Daten und Logik, die durch das neue Tool erreicht werden soll. Dies ist insbesondere mit Blick auf die Anforderungen im medizinischen Bereich sowie die geplante CRM-Integration von strategischer Bedeutung.

Die Entscheidung für ein neues Umfragetool ist daher aus Sicht von Datenschutz, Anpassbarkeit, technischer Flexibilität und langfristiger Systemintegration klar vorteilhaft. Sie bietet nicht nur kurz- bis mittelfristige Verbesserungen in der Durchführung von Umfragen, sondern schafft auch die Grundlage für zukünftige Digitalisierungsschritte innerhalb der TICOH GmbH.

Da die Eigenentwicklung im Rahmen dieses Projekts als Prototyp umgesetzt wurden dient sie vor allem der technischen Validierung und Anforderungsanalyse. Die abschließende Entscheidung über eine dauerhafte Nutzung sei es durch Weiterentwicklung des Prototyps oder Auswahl eines externen Tools wie UmfrageOnline erfolgt zu einem späteren Zeitpunkt im Rahmen einer Make-or-Buy-Bewertung.

3.3 Marktanalyse

Zur fundierten Entscheidungsfindung wurde im Vorfeld des Projekts eine Marktanalyse durchgeführt. Ziel war es, möglich Alternativen zum bisherigen Umfragetool zu identifizieren und hinsichtlich zentraler Kriterien zu vergleichen. In die Analyse einbezogen wurden das bereits verwendete System (QuestionPro), mehrere kommerzielle Umfragesysteme (z.B. UmfrageOnline, EasyFeedback, LimeSurvey, SoSciSurvey) sowie die Option einer Eigenentwicklung.

Neben technischen Merkmalen wie Fragetypen, Bedingungen und Datenschutz wurden auch wirtschaftliche Faktoren wie Lizenzkosten und Erweiterbarkeit berücksichtigt. Die Bewertung erfolgt anhand einer Entscheidungs- und Gewichtungsmatrix, um eine objektive Vergleichbarkeit der Optionen zu gewährleisten.

Die Analyse zeigt, dass insbesondere das Tool UmfrageOnline ein sehr gutes Verhältnis aus Funktionsumfang, Benutzerfreundlichkeit und Kosten aufweist. Die Eigenentwicklung wurde im Rahmen dieses Projekts dennoch verfolgt um zentrale Anforderungen der TICOH GmbH gezielt evaluieren zu können und ein prototypischen Umsetzungsweg aufzuzeigen.

3.4 Entscheidungsgrundlage

Auf Basis der Marktanalyse wurden die relevanten Optionen (bestehendes System, neue kommerzielle Tools und Eigenentwicklung) im Hinblick auf ihre langfristige Nutzbarkeit bewertet. Während das bestehende System (QuestionPro) vergleichsweise hohe Lizenzkosten verursacht und funktional eingeschränkt ist, überzeugten einige neue Tools wie UmfrageOnline mit einem modernen Funktionsspektrum, attraktiven Konditionen und der Möglichkeit, durch eigene Entwicklungen individuelle Erweiterungen vorzunehmen.

Die Entscheidung, im Rahmen dieses Projekts eine Eigenentwicklung umzusetzen, diente primär der Anforderungvalidierung: Die prototypische Umsetzung ermöglichte es, spezifische Anforderungen der TICOH GmbH in einer realen Umgebung zu testen. Somit konnten technische Möglichkeiten ausgelotet und praxisnahe Erfahrungen gesammelt werden.

Langfristig schafft der entwickelte Prototyp eine fundierte Grundlage für eine spätere Make-or-Buy-Entscheidung. Auf Basis der Erkenntnisse kann zukünftig entschieden werden, ob das Umfragetool weiterentwickelt oder durch ein externes System ersetzt wird.

Die in Anhang C dokumentierten Tabellen und Marktvergleiche bildeten die Grundlage für die Entscheidungsfindung im Rahmen dieses Projekts.

3.5 Lastenheft

Auf Basis des zuvor erstellten Lastenhefts wurde ein technisches Umsetzungskonzept für den Prototypen entwickelt.

Dieses Pflichtenheft konkretisiert die technische Architektur, definiert die API-Schnittstellen, beschreibt die Datenhaltung in MySQL sowie die Anzeige und Verarbeitungslogik im Frontend. Die Umsetzung orientierte sich dabei an den im Lastenheft priorisierten Kernanforderungen, und wurde mit Blick auf Modularität und Erweiterbarkeit realisiert. (siehe Anhang D: Pflichtenheft)

4. Entwurfsphase

Nach Abschluss der Analysephase wurde in der Entwurfsphase die technische Umsetzung des geplanten Umfragetools konkretisiert. Ziel der Phase war es, eine modulare Architektur festzulegen, die sowohl eine dynamische Umfragelogik als auch eine datenschutzkonforme Datenverarbeitung ermöglicht. Dabei wurde die Softwarekomponenten definiert, das Datenmodell entworfen und die Gestaltung der Benutzeroberfläche vorbereitet.

Diese Phase bildet die Grundlage für die folgende technische Umsetzungsabschnitte, insbesondere das Architekturdesign, das User Interface sowie das relationale Datenmodell.

4.1 Architekturdesign

Die Architektur des Umfragetools wurde als klassische Client-Server-Anwendung konzipiert. Sie basiert auf einem modularen Aufbau mit klarer Trennung zwischen Frontend, Backend und Datenbank. Ziel dieser Architektur ist eine wartbare, skalierbare und datenschutzkonforme Lösung die sowohl die Anforderungen der TICO GmbH erfüllt als auch zukünftige Erweiterungen (z.B. CRM-Anbindung) berücksichtigt.

Komponentenübersicht

1. Frontend (Client)

Das Frontend wurde als einfache HTML/JavaScript-Anwendung umgesetzt. Es lädt Umfragefragen über eine REST-Schnittstelle im JSON-Format und stellt die dynamisch im Browser dar. Die Logik zur Steuerung der Sichtbarkeit von Fragen (z.B. Anzeige von Folgefragen abhängig von Vorantworten) wird direkt im Client verarbeitet.

2. Backend (Server/API)

Das Backend basiert auf Node.js mit dem Framework Express.js. Es stellt verschiedene REST-Endpunkte zur Verfügung

| API-Endpunkte | Funktion |
|-------------------------|---|
| GET /api/fragen | Alle Umfragefragen aus der Datenbank abrufen |
| POST /api/antworten | Antworten des Nutzers speichern |
| GET /api/status/:token | Gültigkeit und Nutzungsstatus eines Tokens prüfen |
| POST /api/status/:token | Token als „verwendet“ markieren |
| POST /api/status | Neuen Token anlegen |

Tabelle 4.: Endpunkte Server API

3. Datenbank (MySQL)

Die Datenbank dient als zentrale Persistenz Schicht und speichert alle relevanten Informationen in relationalen Tabellen

- fragen (Fragetext, Typ, Optionen, Bedingungen)
- antworten (Nutzerantworten mit Zeitstempel)
- fragebogen_status (Zugriffssteuerung über Tokens)

Die Datenbank ist über vorbereitete SQL-Statements mit dem Backend verbunden. Die Speicherung erfolgt strukturiert und nachvollziehbar, um spätere Auswertung zu ermöglichen.

Kommunikation und Datenfluss

- Das Frontend sendet http-Anfragen an die API-Endpunkte des Express-Backend.
- Das Backend verarbeitet die Anfrage, greift bei Bedarf auf die MySQL-Datenbank zu und liefert eine JSON-Antwort zurück.
- Die gesamte Kommunikation erfolgt über das JSON-Format sowohl beim Abrufen der Fragen als auch beim Übermitteln der Antworten.

Architekturziele

- Modularität: Frontend, Backend und Datenbank sind lose gekoppelt und separat erweiterbar.
- Skalierbarkeit: Die Anwendung kann bei Bedarf um weitere Funktionalitäten erweitert werden (z.B. Ergebnis-Auswertung, Admin-Panel).
- Wartbarkeit: Die saubere Trennung der Schichten erleichtert das Debugging und die Weiterentwicklung.

4.2 Entwurf und Gestaltung der Benutzeroberfläche

Die Benutzeroberfläche (Frontend) des Umfragetools wurde bewusst schlicht und funktional gehalten. Die Oberfläche basiert auf HTML und JavaScript und wird direkt im Browser ausgeführt, ohne dass eine Installation erforderlich ist.

Aufbau der Oberfläche

Beim Aufruf des Umfragetools wird der Nutzer durch die Umfrage geführt eine Frage pro Schritt. Die Oberfläche besteht aus folgenden Hauptelementen:

- Fragetext (z.B. „In welcher Einrichtung arbeiten Sie?“)
- Eingabefeld oder Dropdown Menü je nach Frage typ
- Weiter Button, um zur nächsten sichtbaren Frage zu gelangen

Dynamische Anzeige der Fragen

Ein zentrales Merkmal der Oberfläche ist die dynamische Steuerung der Fragenanzeige. Abhängig von den vorher gegebenen Antworten entscheidet eine Logik im JavaScript-Code, ob eine bestimmte Folgefrage angezeigt wird.

Beispiel:

Wenn in Frage 1 „Zahnarztpraxis“ gewählt wurde, erscheint Frage 2. Andernfalls wird sie übersprungen.

Die Steuerung erfolgt clientseitig und basiert auf dem Attribut anzeigebedingung der Fragenobjekte, die im JSON-Format vom Backend geliefert werden. Die Auswertung erfolgt mit regulären Ausdrücken und Vergleichsoptionen.

Die Gestaltung der Oberfläche folgt gängigen Usability Prinzipien:

- Einfachheit: Keine ablenkenden Elemente oder überflüssige Navigation
- Responsivität: Funktioniert auch auf kleineren Bildschirmen
- Fehlerschutz: Nutzer erhalten Feedback wenn sie versuchen eine Frage ohne Antwort zu Überspringen
- Abschlussanzeige: Nach der letzten Frage erscheint eine Bestätigung

Technischer Hintergrund

Das Frontend lädt die Fragen beim Start über den API-Endpunkt /api/fragen. Alle Fragen liegen im JSON-Format vor und werden bei der Anzeige interpretiert und gerendert. Die Antworten werden am Ende gesammelt (als Objekt antworten) und können in einem späteren Schritt per POST /api/antworten an das Backend gesendet werden.

4.3 Datenmodell

Das Datenmodell des Umfragetools wurde als relationales Datenbankschema in MySQL umgesetzt. Ziel war eine strukturierte, nachvollziehbare und erweiterbare Speicherung aller Umfrageinhalte, Nutzerantworten und Zugriffssteuerungsmechanismen.

1. Fragen

| Feld | Typ | Beschreibung |
|------------------|------------------|---|
| id | INT, Primary Key | Eindeutige technische ID der Frage |
| text | TEXT | Umfrage Text |
| type | VARCHAR(50) | Fragetyp (z.B. „Einfachauswahl“ , „Dropdown“) |
| optionen | TEXT | Antwortoptionen im JSON oder String-Format |
| anzeigebedingung | TEXT (optional) | Bedingung für die Sichtbarkeit (z.B. abhängig von F1) |

Tabelle 5.: fragen Tabelle MySQL

2. Antworten

| Feld | Typ | Beschreibung |
|-------------|------------------|---|
| id | INT, Primary Key | Technische ID |
| frage_id | INT | Fremdschlüssel auf die fragen-Tabelle |
| gruppe_id | INT | Optionale Gruppenzuordnung (z.B. Token-ID) |
| Antworten | TEXT | Die gegebenen Antworten (z.B. Text, Auswahl,Zahl) |
| zeitstempel | DATETIME | Zeitpunkt der Abgabe |

Tabelle 6.: antworten Tabelle MySQL

3. fragebogen_status

| Feld | Typ | Beschreibung |
|---------------|------------------|---|
| id | INT, Primary Key | Technische ID |
| gruppen_token | VARCHAR(255) | Eindeutiger Zugangslink (z.B. „abc123“) |
| genutzt | BOOLEAN | true = bereits verwendet false = offen |
| Zeitpunkt | DATETIME | Zeitpunkt der Einreichung |

Tabelle 7.: fragebogen_status Tabelle MySQL

Beziehung im Modell

- Eine Frage (frage) kann von mehreren Nutzern beantwortet werden 1:n Beziehung zu antworten
- Eine Gruppe (bzw. ein Link) kann mehrere Antworten enthalten 1:n Beziehung zu antworten
- Die Tabelle antworten referenziert sowohl fragen als auch fragebogen_status

Die Struktur ist so ausgelegt, dass sie modular erweitert werden kann -z.B. um weitere Fragetypen, Nutzergruppen, Mehrsprachigkeit oder Auswertungsmetriken.

4.4 Maßnahmen zu Qualitätssicherung

Zur Sicherstellung der Funktionsfähigkeit, Stabilität und Zuverlässigkeit des entwickelten Umfragetools wurden im Verlauf der Entwicklungsphase gezielte Maßnahmen zur Qualitätssicherung durchgeführt. Da es sich um ein sicherheitsrelevantes Projekt im Gesundheitsumfeld handelt, lag besonderes Augenmerk auf Datenschutz, funktionaler Korrektheit und Benutzerfreundlichkeit.

Wie im Abschnitt 2.3 (Entwicklungsprozess) beschrieben, wurde die Software iterativ entwickelt. Regelmäßige Abstimmungen mit dem Fachbetreuer bei der TICOH GmbH ermöglichten frühzeitiges Feedback, wodurch potenzielle Fehlerquellen bereits während der Entwicklung erkannt und behoben werden konnten.

Durchgeführte Qualitätssicherungsmaßnahmen:

1. Funktionstests

Alle wesentlichen Funktionen des Umfragetools wurden einzeln und in Kombination getestet. Dazu zählten u.a.:

- Abruf der Fragen über die API (GET /api/fragen)
- Speicherung von Antworten (POST /api/antworten)
- Prüfung von Zugriffs-Token (GET /api/status/:token)
- Markierung eines Tokens als „genutzt“ (POST /api/status/:token)

Die Tests erfolgten mithilfe von Postman und direkt über das browserbasierte Frontend.

2. Fehlertests und Validierung

Es wurden gezielt Tests mit ungültigen oder unvollständigen Eingaben durchgeführt, um sicherzustellen, dass das System korrekt reagiert (z.B. mit HTTP 400 bei fehlenden Feldern). Ebenso wurde getestet, ob bereits verwendete Tokens zuverlässig blockiert werden.

3. Usability-Tests

Die Benutzerführung wurde mit Testpersonen aus dem Umfeld simuliert, um Rückmeldung zur Verständlichkeit und Bedienbarkeit zu erhalten. Dabei wurde überprüft, ob die Fragen logisch angezeigt werden und ob die Eingabeformulare verständlich und funktional sind.

4. Code-Review

vor der Finalisierung der Backend-Komponenten wurden die wichtigsten Routen, Funktionen und Datenbankabfragen gemeinsam mit dem Fachbetreuer gesichtet. Dabei wurden Best Practices, Sicherheit, Lesbarkeit und Erweiterbarkeit des Codes überprüft.

5. Dokumentation und Nachvollziehbarkeit

alle API-Routen, Datenbanktabellen und Konfigurationen wurden strukturiert dokumentiert, um eine spätere Wartung zu erleichtern. Besonderes Augenmerk lag auf der Nachvollziehbarkeit der Fragenlogik und der Datenflüsse.

Diese Maßnahmen bilden die Grundlage für ein robustes, nachvollziehbares und technisch sicheres Umfragetool. Weitere Details zur Testdurchführung und konkreten Beispielen von Testaufrufen und Fehlersimulationen folgen.

4.5 Pflichtenheft

Nach Abschluss der Entwurfsphase wurde für das Umfragetool der TICOH GmbH ein Pflichtenheft erstellt. Dieses Dokument basiert auf dem Lastenheft aus Abschnitt 3.4 und beschreibt detailliert, wie die Anforderungen technisch umgesetzt werden sollen. Es umfasst spezifische Angaben zur Systemarchitektur, den verwendeten Technologien, Schnittstellen und Sicherheitsaspekte. (Siehe Anhang E Pflichtenheft)

5. Implementierungsphase

Die Technische Umsetzung des Umfragetools für medizinische Einrichtungen wird in diesem Abschnitt detailliert beschrieben. Für die Entwicklung und das Testen der Anwendung wurde Visual Studio Code als Entwicklungsumgebung eingesetzt. VS Code bot eine übersichtliche und effiziente Plattform für das Schreiben Testen und Verwalten des gesamten Codes.

Die Programmierung erfolgt mit JavaScript, unter Einsatz von Node.js und dem Express.js-Framework für das Backend. Express eignete sich hervorragend für die Erstellung der REST-API, über die Daten zwischen Frontend und Backend ausgetauscht wurden. Als Datenbank kam MySQL zum Einsatz. Die Verwaltung der Datenbank erfolgt mithilfe von pypMyAdmin.

Zu Beginn des Projekts wurde Entwicklungsumgebung eingerichtet, einschließlich der Installation aller notwendigen Node.js-Pakete (z.B. express, mysql2, cors, dotenv). Die Verzeichnisstruktur wurde so gewählt, dass Frontend-Dateien im Ordner public liegen und API-Routen modular im Ordner routes abgelegt sind.

Im Fokus der technischen Umsetzung stand:

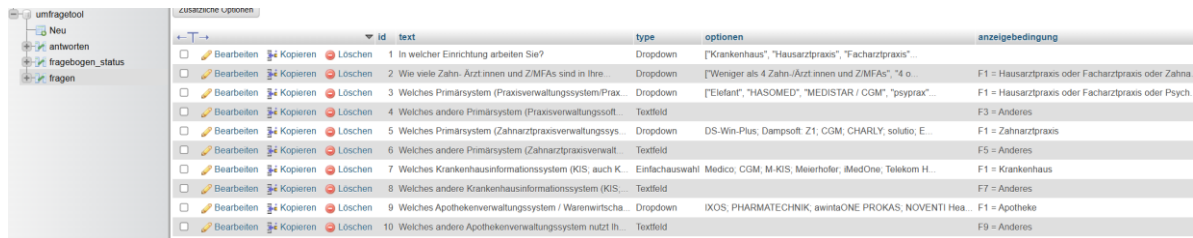
- Die dynamische Anzeige der Umfragefragen, inklusive Sichtbarkeitslogik je nach vorheriger Antwort,
- Die Speicherung von Antworten in der Datenbank, gruppiert nach Einreichungen,
- Die Entwicklung eines optionalen Token-Systems zur einmaligen Teilnahme,
- Sowie eine Admin-Ansicht, über die abgeschlossene Umfragen eingesehen werden können.

Für die eindeutige Zuordnung von Einreichungen wurde eine UUID-basierte Einreichungs-ID verwendet. Diese erlaubt eine strukturierte Auswertung und spätere Nachvollziehbarkeit pro beantworteten Fragebogen. Auch wurde eine Logik zur Prüfung und Markierung von verwendetem Token umgesetzt.

Das Umfragetool wurde regelmäßig getestet, sowohl im Browser (Frontend) als auch über Postman (Backend-API), um eine korrekte Funktion und Einhaltung der Datenschutzanforderung sicherzustellen.

5.1 Implementierung der Datenstrukturen

Die Datenstrukturen des Umfragetools wurden auf Basis eines relationalen Datenbankmodells mit MySQL umgesetzt. Die Tabellenstruktur wurde so entworfen, dass sie sowohl die Fragenlogik, die Speicherung der Teilnehmerantworten als auch die Verwaltung optionaler Zugangstoken abbildet.



| | id | text | type | optionen | anzeigebedingung |
|--|----|--|----------------|---|---|
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 1 | In welcher Einrichtung arbeiten Sie? | Dropdown | ["Krankenhaus", "Hausarztpraxis", "Facharztpraxis"] | |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 2 | Wie viele Zahn-/Ärzt:innen und Z/MFAs sind in Ihre... | Dropdown | ["Weniger als 4 Zahn-/Ärzt:innen und Z/MFAs", "4 o... | F1 = Hausarztpraxis oder Facharztpraxis oder Zahna... |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 3 | Welches Primärsystem (Praxisverwaltungssystem) Prax... | Dropdown | ["Elefant", "HASOMED", "MEDISTAR / CGM", "psypax"] | F1 = Hausarztpraxis oder Facharztpraxis oder Psych... |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 4 | Welches andere Primärsystem (Praxisverwaltungssoft... | Textfield | | F3 = Anderes |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 5 | Welches Primärsystem (Zahnarztpraxisverwaltungsys... | Dropdown | DS-Win-Plus; Dampsoft Z1; CGM, CHARLY; solutio; E... | F1 = Zahnarztpraxis |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 6 | Welches andere Primärsystem (Zahnarztpraxisverwal... | Textfield | | F5 = Anderes |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 7 | Welches Krankenhausinformationssystem (KIS, auch K... | Einfachauswahl | Medico; CGM; M-KIS; Meierhofer; MedOne; Telekom H... | F1 = Krankenhaus |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 8 | Welches andere Krankenhausinformationssystem (KIS... | Textfield | | F7 = Anderes |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 9 | Welches Apothekenverwaltungssystem / Warenwirtsch... | Dropdown | IXOS; PHARMATECHNIK; awintaONE PROKAS; NOVENTI Hea... | F1 = Apotheke |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 10 | Welches andere Apothekenverwaltungssystem nutzt ih... | Textfield | | F9 = Anderes |

Abbildung 1.: Struktur der Datenbanktabelle im Umfragetool (PHPMAdmin).

Zentrale Datenbanktabelle:

- fragen
Diese Tabelle enthält alle Umfragefragen, inklusiv Typ (z.B. Einfachauswahl, Freitext) Fragenformulierung und optionalen Antwortoptionen. Zudem ist für jede Frage eine Anzeigebedingung definierbar, mit der dynamische Fragenverläufe realisiert werden können.
- antworten
In dieser Tabelle werden alle gegebenen Antworten gespeichert. Jede Antwort ist einer `frage_id` zugeordnet und gehört zu einer bestimmten Einrichtung (`einrichtung_id`). Zusätzlich wird gespeichert, wann die Antwort abgegeben wurde.
- fragebogen_status
Diese Tabelle wird verwendet, um optional die Teilnahme über personalisierte Zugangstoken zu kontrollieren. Jeder Token ist eindeutig, kann nur einmal verwendet werden und enthält eine zugehörige E-Mail-Adresse. Über das Feld genutzt und einen Zeitstempel wird sichergestellt, dass Tokens nicht mehrfach eingelöst werden können.

Die Speicherung der Einreichungen erfolgt gruppiert über eine automatisch generierte UUID (`einreichung_id`). Diese wird bei jeder Abgabe erzeugt und ermöglicht es alle Antworten einer Person oder Sitzung logisch zusammenzufassen. Unabhängig davon, ob ein Token verwendet wurde.



| | id | frage_id | gruppe_id | antwort | zeitstempel | einreichung_id |
|--|----|----------|-----------|---|---------------------|--------------------------------------|
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 1 | 1 | 0 | Krankenhaus | 2025-05-20 17:18:11 | fe5e0d74-cf4e-4bf6-8dd9-cc0d37233012 |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 2 | 7 | 0 | Medico | 2025-05-20 17:18:11 | fe5e0d74-cf4e-4bf6-8dd9-cc0d37233012 |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 3 | 1 | 354 | Krankenhaus | 2025-05-20 17:18:40 | 286b9cb1-6808-46c8-aa44-1c51a49441c9 |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 4 | 7 | 354 | Medico | 2025-05-20 17:18:40 | 286b9cb1-6808-46c8-aa44-1c51a49441c9 |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 5 | 1 | 3503034 | Hausarztpraxis | 2025-05-20 17:19:02 | 56b0723e-d64e-4dce-b26c-e93c7f0cc506 |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 6 | 2 | 3503034 | Weniger als 4 Zahn-/Ärzt:innen und Z/MFAs | 2025-05-20 17:19:02 | 56b0723e-d64e-4dce-b26c-e93c7f0cc506 |
| <input type="checkbox"/> Bearbeiten <input type="checkbox"/> Kopieren <input type="checkbox"/> Löschen | 7 | 3 | 3503034 | Elefant | 2025-05-20 17:19:02 | 56b0723e-d64e-4dce-b26c-e93c7f0cc506 |

Abbildung 2.: Teilnehmerantworten, Gruppenzuordnung und Zeitstempel (PHPMAdmin).

Die Datenbankverbindung wurde über das `mysql2`-Modul realisiert. SQL-Abfragen wurden serverseitig in Express.js-Routen implementiert. Besonderer Fokus lag darauf alle Benutzer Inputs sicher und valide zu verarbeiten, um Fehler zu vermeiden.

5.2 Dynamische Umfrageoberfläche mit Anzeige-Logik

Die Benutzeroberfläche des Umfragetools wurde mit HTM und JavaScript umgesetzt und so gestaltet das sowohl die Nutzerfreundlichkeit als auch die technische Funktionalität im Fokus steht. Die HTML-Dateien wurden im Verzeichnis /public abgelegt und direkt vom Express-Server aufgeliefert.

```
5 const path = require('path');
6 const app = express();
7
8 app.use(cors());
9 app.use(express.json());
10 app.use(express.static(path.join(__dirname, 'public')));
```

Abbildung 3.: Konfiguration des Express-Servers (VS Code).

Im Zentrum steht die umfrage.html, welche dynamisch Fragen vom Server über die Route GET /api/fragen lädt und mithilfe von JavaScript auf der Seite darstellt. Die Fragen erscheinen abhängig von definierten Anzeigebedingungen. Dabei wurde eine eigene Anzeige-Logik entwickelt, um bedingte Verzweigungen im Fragenverlauf zu ermöglichen.

```
1 const express = require("express");
2 const router = express.Router();
3 const db = require("../db");
4
5
6 router.get("/", (req, res) => {
7   const sql = "SELECT * FROM fragen ORDER BY id ASC";
8   db.query(sql, (err, results) => {
```

Abbildung 4.: Backend-Route zur Bereitstellung der Umfragefragen (VS Code).

Ein „Weiter“-Button steuert den Fragenfortschritt. Am Ende werden die eingegebenen Antworten gesammelt, in ein JSON-Format überführt und über die Route POST /api/antworten an das Backend übermittelt. Wurde die Umfrage über einen personalisierten Token-Link geöffnet wird dieser beim Absenden als verwendet markiert.

```
const einreichung_id = randomUUID();

const werte = Object.entries(antworten).map(([frage_id, antwort]) => [
  parseInt(frage_id),
  gruppe_id || null,
  antwort,
  new Date(),
  einreichung_id
]);

const sql = `
  INSERT INTO antworten (frage_id, gruppe_id, antwort, zeitstempel, einreichung_id)
  VALUES ?
`;
```

Abbildung 5.: Verarbeitung der übermittelten Antworten zur Speicherung in Datenbank.(VS Code).

Neben der Umfrageseite wurden zusätzlich Administrationsoberflächen entwickelt:

- admin.html: zeigt strukturierte Umfrageeinreichungen samt Antworten, Zeitstempel und zugehörigen Token Informationen
- token_admin.html: ermöglicht das manuelle Erstellen, Anzeigen und Kontrollieren von Zugangstokens.
- verteilen.html: unterstützt das Generieren und Verteilen von Umfragelinks, wahlweise mit einmaligem Zugang über Token.

Durch saubere Trennung von Frontend und Backend sowie modulare Gestaltung der HTML-Seiten konnte eine wartbare und erweiterbare Oberfläche realisiert werden.

Zur Veranschaulichung der Benutzeroberfläche und ihrer Funktionsweise sind exemplarische Screenshots des Prototyp Umfragetools sowie der Administrationsansicht im Anhang dokumentiert. Die Dargestellte Seite wird über einen Token basierten Zugangslink aufgerufen. Die dynamisch geladenen

Fragen erscheinen gemäß der im Backend hinterlegten Anzeigelogik und können direkt beantwortet werden. (Siehe Anhang F)

5.3 Backend Logik und Schnittstellen

Die API-Routen wurden in separaten Modulen für Fragen, Antworten und Zugangs Status gekapselt und im Hauptserver über `app.use()` eingebunden. Alle Schnittstellen folgen dem REST-Prinzip und verwenden das JSON-Format für Anfragen und Antworten.

```
9 app.use(express.json());
10 app.use(express.static(path.join(__dirname, 'public')));
11
12 // Routen
13 const fragenRoute = require('./routes/fragen');
14 app.use('/api/fragen', fragenRoute);
15
16 const antwortenRoute = require('./routes/antworten');
17 app.use('/api/antworten', antwortenRoute);
18
19 const statusRoute = require('./routes/status');
20 app.use('/api/status', statusRoute);
21
```

Abbildung 6.: Einbindung der modularen API-Routen im Express-Backend (VS Code).

Die GET-Route `/api/fragen` stellt die Umfragefragen aus der Datenbank bereit, während über die POST-Route `/api/antworten` alle Antworten an den Server übermittelt werden.

```
15 const einreichung_id = randomUUID();
16
17 const werte = Object.entries(antworten).map(([frage_id, antwort]) => {
18   parseInt(frage_id),
19   gruppe_id || null,
20   antwort,
21   new Date(),
22   einreichung_id
23 });
24
25 const sql = `
26   INSERT INTO antworten (frage_id, gruppe_id, antwort, zeitstempel, einreichung_id)
27   VALUES ?
28 `;
```

Abbildung 7.: Serverseitige Verarbeitung und Speicherung der Umfrageantworten (VS Code).

Diese Antworten werden clientseitig in einem JSON-Objekt gesammelt und mit einem eindeutigen `einreichung_id` (UUID) gruppiert in der Datenbank gespeichert. Jede Antwort enthält zusätzlich die zugehörige `frage_id`, die tatsächliche Eingabe (`antwort`), einen Zeitstempel und optional die `gruppe_id` des verwendeten Tokens.

Wenn ein Umfragelink mit Token geöffnet wird, überprüft das Backend zunächst ob der Token gültig und noch nicht verwendet wurde. Nach dem Absenden der Umfrage wird dieser als verwendet markiert. So wird verhindert, dass sensible Umfragen mehrfach beantwortet werden können.

```
9 const sql = `
10   SELECT genutzt
11   FROM fragebogen_status
12   WHERE gruppen_token = ?
13 `;
14
15 db.query(sql, [token], (err, results) => {
16   if (err) {
17     console.error("Fehler bei der Statusabfrage:", err);
18     return res.status(500).json({ error: "Interner Serverfehler" });
19   }
20
21   if (results.length === 0) {
22     return res.json({ gültig: false });
23   }
24
25   const genutzt = results[0].genutzt;
26   res.json({ gültig: true, genutzt });
27 });
28
```

Abbildung 8.: Überprüfen des Tokens auf Gültigkeit und Nutzung vor Umfragestart (VS Code).

Zur Absicherung der Anwendung werden ausschließlich vorbereitete SQL-Statements verwendet. Alle API-Routen sind klar definiert und kapseln die Logik sauber vom Datenzugriff ab. Dadurch wird das Risiko von SQL-Injection deutlich reduziert. Gleichzeitig sorgt die klare Trennung von Logik und Datenhaltung für eine bessere Wartbarkeit und erhöhte Sicherheit des Gesamtsystems.

6. Projektabschluss

Im abschließenden Kapitel wird die praktische Umsetzung des Umfragetools zusammengefasst und reflektiert. Es werden die erreichten Ziele bewertet sowie ein Ausblick auf mögliche Weiterentwicklungen gegeben.

6.1 Zusammenfassung der Umsetzung

Im Rahmen des Projekts wurde eine datenschutzkonformes Umfragetool für medizinische Einrichtungen entwickelt. Ziel war es, ein flexibles, modular erweiterbares System zu schaffen, das sowohl interne Mitarbeiterumfragen als auch externe Kundenfeedbacks datenschutzgerecht abbildet. Die Anwendung wurde mit Node.js und Express.js als Backend sowie mit HTML und JavaScript im Frontend realisiert. MySQL diente als Datenbank zur strukturierten Speicherung der Fragen, Antworten und Tokens. Die Fragenlogik wurde dynamisch umgesetzt, bedingte Fragenanzeige sowie ein optionales Token-System zur Einmalverwendung von Umfragen vollständig integriert.

6.2 Test und Qualitätssicherung

Zur Sicherstellung der Funktionsfähigkeit wurden regelmäßig manuelle Tests durchgeführt. Dabei wurde insbesondere auf folgende Aspekte geachtet:

- Datenfluss: Alle eingegeben Antworten wurden korrekt als JSON an das Backend übermittelt und in der Datenbank gespeichert.
- Fragenlogik: Die Anzeige bedingter Fragen funktionierte in allen Testfällen erwartungsgemäß.
- Token-Prüfung: Bei aktivierter Token-Funktion wurden Mehrfacheinreichungen erfolgreich verhindert.
- Fehlerszenarien: Auch bei ungültigen Tokens, leerer Eingabe oder Serverfehlern reagiert das System stabil und benutzerfreundlich.

Zusätzlich wurde der Quellcode durch manuelles Code-Review durch den Projektbetreuer auf Verständlichkeit und Fehleranfälligkeit überprüft.

6.3 Lessons Learned

Im Verlauf des Projekts konnten vielfältige praktische Erfahrungen gesammelt werden, die für zukünftige Softwareentwicklungen von großem Nutzen sein werden. Besonders deutlich wurde die Bedeutung eines Modularens Aufbaus. Durch die saubere Trennung von Routen, Views und Funktionseinheiten wie der Token Verwaltung ließ sich das Projekt flexibel erweitern und strukturieren.

Auch die frühzeitige Planung der Fragenlogik erwies sich als essenziell. Eine gut durchdachte Struktur für bedingte Fragen sparte im späteren Verlauf viel Entwicklungsaufwand und half logische Fehler zu Vermeiden.

Ein weiterer zentraler Lernaspekt war die Relevanz konsequenter Tests: Durch wiederholte Tests in unterschiedlichen Szenarien konnten zahlreiche kleinere, aber entscheidende Fehler frühzeitig erkannt und behoben werden.

Nicht zuletzt zeigte sich, dass eine stabile und durchdachte Datenstruktur die Grundlage für eine funktionierende Benutzeroberfläche bildet. Insbesondere im Adminbereich, wo die Darstellung der Einreichungen direkt auf der Datenbanklogik aufbaut.

6.4 Ausblick

Mit der Erfolgreichen Umsetzung der Kernfunktionen bildet der entwickelte Prototyp eine solide Grundlage, um zentrale Anforderungen an ein Umfragetool in der Praxis zu evaluieren.

Mögliche zukünftige Erweiterungen auf Basis des Prototyps könnten u.a. die Einführung von Filter und Exportfunktionen im Adminbereich, die Verwaltung von Nutzerrollen sowie die Integration eines automatisierten E-Mail-Versands zur effizienten Verteilung der Umfragelinks beinhalten.

Auch eine Mehrsprachigkeit oder barrierefreie Gestaltung wären denkbar Weiterentwicklungen, sollten diese Anforderungen im Rahmen einer späteren Toolauswahl oder Eigenentwicklung relevant werden.

7. Danksagung

Mein besonderer Dank gilt meinem Praktikumsbetrieb für die Möglichkeit dieses Projekt praxisnah umzusetzen. Ebenso danke ich meinem Ausbilder und den fachlichen Ansprechpartnern für die konstruktive Begleitung.

Bei der technischen Umsetzung habe ich zusätzlich frei verfügbare Fachquellen sowie unterstützende Tools und Testumgebungen genutzt, um einzelne Aspekte zu überprüfen, zu strukturieren oder textlich zu verfeinern. Die gesamte Konzeption und Implementierung des Projekts erfolgte eigenverantwortlich!

Anhang

A. Detaillierte Zeitplanung der Projektphasen

| Phase | Aufgabe | 6 |
|------------------------------|---|-----------|
| Analysephase | Erarbeitung des Projektziels, Zielgruppe und Einsatzrahmen | 3 |
| Analysephase | Strukturierung der Fragenlogik mit Anzeige-/Verzweigungsbedingungen | 4 |
| Analysephase | Erstellung Lasten- und Pflichtenheft | 3 |
| Summe Analysephase | | 10 |
| Entwurfsphase | Entwurf der JSON-Fragenstruktur (Frageobjekte, Anzeige-Bedingungen, Optionen) | 3 |
| Entwurfsphase | Modellierung des Datenbankschemas (MySQL) mit UUID-Einreichungskonzept | 4 |
| Entwurfsphase | Konzeption der API-Routenstruktur (GET/POST /api/fragen, /antworten, /status) | 3 |
| Summe Entwurf | | 10 |
| Implementierungsphase | Anbindung der Datenbank an Express.js und Test der Verbindung | 5 |
| Implementierungsphase | Implementierung der API-Endpunkte und Speicherung der Antworten | 10 |
| Implementierungsphase | Einführung der Token-Validierung und Prüfung auf Einmaligkeit | 7 |
| Implementierungsphase | Implementierung der dynamischen Anzeige-Logik (JS) | 8 |
| Implementierungsphase | Entwicklung des Adminbereichs inkl. Tokenverwaltung und Antwortanzeige | 10 |
| Summe Implementierung | | 40 |
| Dokumentationsphase | Verfassen der Projektdokumentation inkl. technischer Kapitel | 10 |
| Dokumentationsphase | Einbindung von Abbildungen, Tabellen und Screenshots | 6 |
| Dokumentationsphase | Erstellung des Abbildungsverzeichnisses, Glossars und Inhaltsverzeichnisses | 4 |
| Summe Dokumentation | | 20 |
| Summe Gesamt | | 80 |

Abbildung 9.: Detaillierte Zeitplanung Projektphasen (Excel)

B. Beispiel Use Case Diagramm

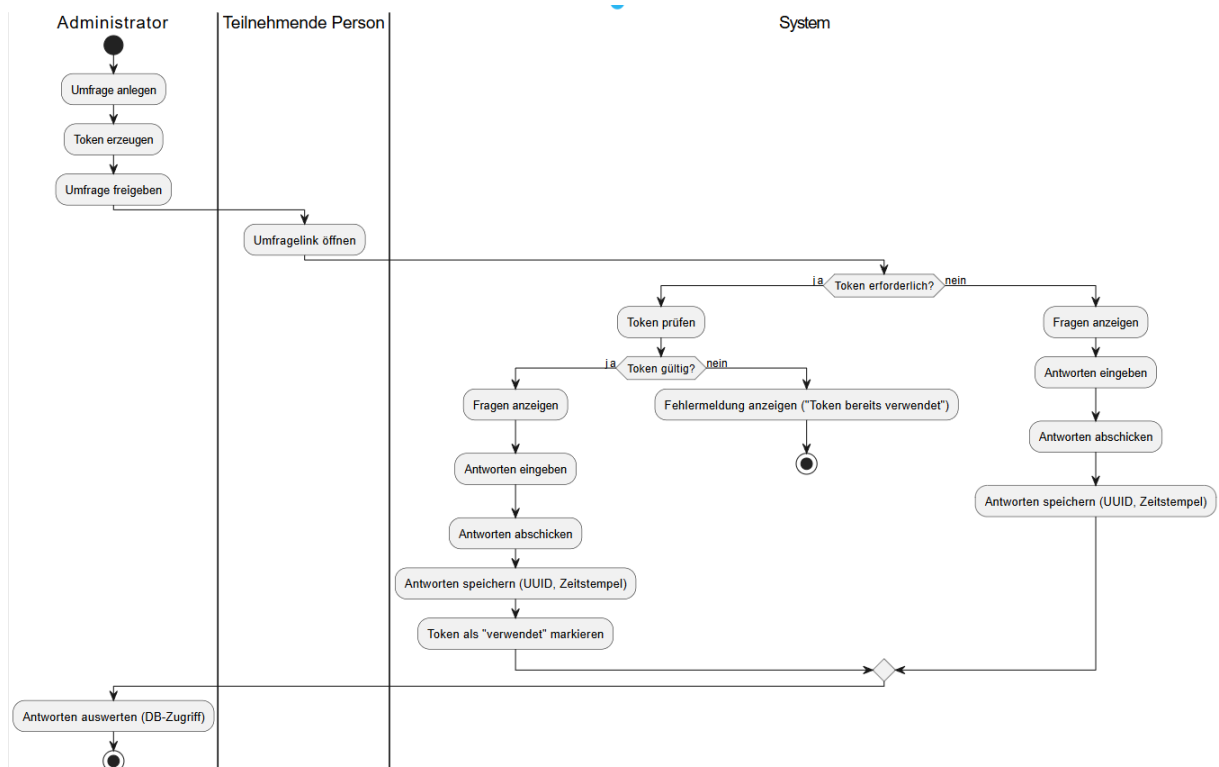


Abbildung 10.: Anwendungsfalldiagramm - Umfragetool (draw.io)

C. Marktanalyse und Entscheidungsmatrix

| Tool | Preis | Vorteile | Nachteile |
|-----------------------------|--|---|--|
| Lime-survey | <ul style="list-style-type: none"> • Experte: 75€/Monat (jährliche Abrechnung) • 5-10 Admins Pricing | | <ul style="list-style-type: none"> • Unübersichtliche Fragensettings und Logikeneinstellung (keine Testversion möglich) • Formatierung der Antworttexte nicht möglich |
| Easy Feedback | <ul style="list-style-type: none"> • Starter: 49€/Monat oder 490€/Jahr • 1 Account Umfrage-Tarif bestellen | <ul style="list-style-type: none"> • Intuitives Tool • Erstellen von Teilnehmerlisten für Versand an Gruppen (ähnlich wie Communities in QP) • Umfragelink kann teilweise angepasst werden • Email-Einladungssystem + Reminder • SPSS Export • Darstellung der Auswertung in Diagrammen | <ul style="list-style-type: none"> • Für uns relevante Funktionen sind alle mit „Starter“ verfügbar, mehrere Benutzerlizenzen erst ab „Business“ (948€ + zusätzliche Kosten/Account/ Jahr) • Logiken funktionieren nur über Seiten hinweg, nicht innerhalb einer Seite • Fragentext kann nicht formatiert werden (fett, kursiv, etc.) • Bei Abfrage von Zahlen kann nicht vorgegeben werden was min und <u>max</u> Wert ist, nur dass es eine Zahl sein muss • Email Versand: Kann als Spam eingestuft werden |

Abbildung 11.: Marktanalyse 1

| | | | |
|-----------------------|---|---|---|
| Umfrage online | <ul style="list-style-type: none"> • Business: 45€/ Monat oder 540€/ Jahr • 3 Zugänge (weitere Admin Accounts können dazu gekauft werden, Kosten je nach Abrechnung unterschiedlich) Verwalten Sie Ihr Abonnement | <ul style="list-style-type: none"> • Link kann teilweise angepasst werden • E-Mail-Versand mit Serienbriefelementen (Reminder) • Intuitives Aufbauen der Umfragen und Logiken • Antworttext kann formatiert werden • Darstellung der Auswertung durch Diagramme • Durch Eigenentwicklung erweiterbar | <ul style="list-style-type: none"> • Kein Download von SPSS Daten (aber csv → kompatibel mit SPSS) • Es können keine Verteilerlisten hinterlegt werden • Keine Übersicht über eingesetzte Logiken • Bei Abfrage von Zahlen kann nicht vorgegeben werden was min und <u>max</u> Wert ist, nur dass es eine Zahl sein <u>muss</u> |
| SoSci | Variable nach gebuchtem Plan (abhängig von Zeitraum und Stichprobengröße) Preise | <ul style="list-style-type: none"> • Komplett anpassbar • Flexibles Buchen der Funktionen • Download als SPSS/ Excel Daten | <ul style="list-style-type: none"> • Umständliches Design der Fragen • Keine Darstellung der Auswertung wird angezeigt |

Abbildung 12.: Marktanalyse 2

| Bewertungsmatrix zu Entscheidungsfindung: - bestehendes System - neues System - Eigenentwicklung | | Kosten | Bewertung Vorteile | Bewertung Nachteile | Bewertung Erweiterbarkeit |
|---|--|--------------------|--------------------|---------------------|---------------------------|
| Option 1 - Bestehendes System | | | ++ | -- | -- |
| Question Pro | | | | | |
| Angebot Jahrlizenz, pro Jahr | | 10.115,00 € | | | |
| Option 2 - neues System | | | | -- | + |
| Lime-survey | | | | | |
| Angebot Monatslizenz | | 29,00 € | | | |
| Jährlich Kosten | | 348,00 € | | | |
| Easy Feedback | | | ++ | -- | - |
| Angebot Monatslizenz | | 49,00 € | | | |
| Jährlich Kosten | | 490,00 € | | | |
| Umfrage online | | | +++ | - | +++ |
| Angebot Monatslizenz | | 45,00 € | | | |
| Jährlich Kosten | | 540,00 € | | | |
| SoSci | | | + | -- | ++ |
| Jährlich Kosten min. | | 399,00 € | | | |
| Option 3 - Eigenentwicklung | | | ++ | -- | +++ |
| Anforderungsanalyse, Aufwand in Stunden | | 160Std. | | | |
| Kosten je Stunde | | 22,83 € | | | |
| Kosten | | 3.652,80 € | | | |
| Entwicklung, Aufwand in Stunden | | 460Std. | | | |
| Kosten je Stunde | | 22,83 € | | | |
| Kosten | | 10.501,80 € | | | |
| Kosten gesamt | | 14.154,60 € | | | |

Abbildung 13.: Bewertungsmatrix (Excel)

D. Auszug Lastenheft

1. Zielbestimmung

Ziel des Projekts ist die Entwicklung eines webbasierten Umfragetools zur strukturierten Erfassung von Rückmeldungen in medizinischen Einrichtungen. Das Tool soll datenschutzkonform nach DSGVO-Vorgaben umgesetzt werden, eine flexible Fragenlogik unterstützen und eine einfache Auswertung ermöglichen. Die Lösung richtet sich an interne Fachabteilungen sowie externe Projektpartner im Rahmen von Digitalisierungsprojekten.

2. Produkteinsatz

Das Umfragetool wird von der TICOH GmbH im Rahmen von Pilotprojekten eingesetzt, insbesondere zur Erhebung von Erfahrungen mit KIM, eRezept und anderen TI-Diensten. Die Anwendung soll sowohl auf Desktop- als auch auf mobilen Endgeräten im Browser nutzbar sein.

3. Anforderungen aus Sicht der Auftraggeber

- Das Tool muss datenschutzkonform arbeiten (z. B. durch anonymisierte Erhebung und einmalige Tokens).
- Es soll eine flexible Fragenverwaltung ermöglichen (inkl. bedingter Anzeige).
- Ergebnisse sollen zentral gespeichert und im Adminbereich strukturiert dargestellt werden.
- Zugangscodes (Tokens) sollen optional nutzbar sein.
- Die Lösung muss wartbar und modular aufgebaut sein.

4. Nicht-funktionale Anforderungen

- Die Bedienung soll einfach, responsiv und barrierearm sein.
- Es ist kein Login für Teilnehmer erforderlich.
- Der Code soll gut dokumentiert, modular und nachvollziehbar aufgebaut sein.
- Der Quellcode wird intern bei TICOH gepflegt.

E. Auszug Pflichtenheft

1. Zielsetzung und Umsetzungsidee

Auf Basis des Lastenhefts wurde ein technisches Konzept entwickelt, das eine Umsetzung mit HTML, CSS/JS im Frontend sowie Node.js mit Express.js und MySQL im Backend vorsieht. Die Anwendung folgt einem klaren Client-Server-Prinzip mit RESTful-API-Struktur und nutzt serverseitige Validierung zur Absicherung der Datenintegrität.

2. Funktionale Anforderungen

- Fragen werden über die Route `/api/fragen` bereitgestellt. Die Anzeige erfolgt dynamisch über JavaScript – abhängig von definierten Anzeigebedingungen.
- Beim Absenden überträgt das Frontend die Antworten im JSON-Format an die Route `/api/antworten`.
- Im Backend werden die Antworten samt Zeitstempel und UUID gespeichert. Optional wird eine `gruppe_id` übermittelt (z. B. bei Tokenzugang).
- Der Token wird vor Start geprüft und bei Abgabe als „verwendet“ markiert. Die zugehörigen Routen sind `/api/status/:token` (GET) und `/api/status/:token` (POST).

3. Technische Umsetzung

- Die HTML-Dateien liegen im /public-Verzeichnis und werden statisch über Express ausgeliefert.
- Die Routen für Fragen, Antworten und Status sind modular gekapselt.
- Daten werden in einer MySQL-Datenbank in vier Tabellen verwaltet (fragen, antworten, fragebogen_status, einreichung_id).
- Eingaben werden im Backend mit vorbereiteten SQL-Statements verarbeitet.

4. Qualität und Wartbarkeit

- Getrennte Verantwortlichkeiten (Fragenlogik, Tokenvalidierung, Datenhaltung) ermöglichen eine hohe Wartbarkeit.
- Die Umsetzung ist modular (jede Route ein Modul), Erweiterungen (z. B. GUI, Export) sind vorgesehen.
- Die Anwendung wurde kontinuierlich getestet (z. B. Token-Verbrauch, Anzeigeverzweigungen).

F. Beispielhafte GUI-Ansicht des Prototyps

← ↻ 📄 localhost:3000/umfrage.html?token=64a40b04-f438-42c6-a058-f96ec5705728

Fragebogen

Welches andere Krankenhausinformationssystem (KIS; auch Krankenhaus-Software) nutzt Ihre Einrichtung?

TestingKrankenhausinfo.

Weiter

Abbildung 14.: Darstellung einer Umfragefrage mit Token basiertem Zugriff (Prototypansicht).

Einreichung

einreichung_id: 52ae3968-8ab5-49b3-b9a1-4c74c55bb237 | gruppe_id: 64 | E-Mail: test@testing.com

| Frage-ID | Frage text | Antwort |
|----------|---|-------------------------|
| 1 | In welcher Einrichtung arbeiten Sie? | Krankenhaus |
| 7 | Welches Krankenhausinformationssystem (KIS; auch Krankenhaus-Software) nutzt Ihre Einrichtung? | Anderes |
| 8 | Welches andere Krankenhausinformationssystem (KIS; auch Krankenhaus-Software) nutzt Ihre Einrichtung? | TestingKrankenhausinfo. |

Abbildung 15.: Übersicht einer Umfrageeinreichung mit zugehörigem Token im Adminbereich

G. Projektabnahme

Projektname:

Entwicklung eines prototypischen Umfragetools-

Prüfling:

Lewczuk, Justin

Ausbildungsbetrieb:

TICOH GmbH, Klinikweg 23, 22081 Hamburg

Projektzeitraum:

12.05.2025 – 23.05.2025

Projektbetreuer:

Markus Habetha

Abnahmedatum:

23.05.2025

Kurzbeschreibung:

Im Rahmen dieses Projekts wurde ein webbasiertes Umfragetool für die TICOH GmbH prototypisch entwickelt. Ziel war die Umsetzung zentraler Anforderungen hinsichtlich Fragenlogik, Datenschutz, Token-basierter Zugriffskontrolle und Backend-Anbindung. Das Tool bildet die Grundlage für eine spätere Make-or-Buy-Entscheidung.

Abnahmeerklärung:

Das Projekt wurde abgeschlossen und an den Projektbetreuer übergeben. Die im Rahmen der Anforderungsanalyse definierten Kernfunktionen wurden vollständig umgesetzt. Das Projekt entspricht den Erwartungen an eine prototypische Lösung im Rahmen der betrieblichen Aufgabenstellung.


Bemerkungen :

Die finale visuelle Gestaltung (Design und CSS-Styling) erfolgt ggf. nach der Dokumentationsabgabe. Eine Weiterentwicklung des Prototyps wird intern geprüft.


Ort, Datum

Hamburg, den 23.05.2025

Unterschrift Prüfling


Justin Lewczuk

Unterschrift Projektbetreuer


M. Habetha