Abschlussprojekt in der Informatikausbildung GymInf

"Einmal um die Schweiz"

Eine Webplattform für Sportliche Challenges in und ausserhalb der Schule.

Inhaltsverzeichnis

Einleitung

1.1 Hintergrund des Projekts

In der heutigen Arbeitswelt sind Herausforderungen wie jene, die durch Plattformen wie "bike-to-work" geboten werden, zu einer festen Größe geworden. Obwohl es mit "bike-to-school" eine ähnliche Aktion für Schüler gibt, ist diese weniger in den Schulalltag integriert und findet vor allem auf gymnasialer Stufe kaum Beachtung. Beide Aktionen legen den Schwerpunkt auf das bloße Mitmachen. Mein Projekt hingegen betont stärker den Aspekt der Herausforderung und des Wettbewerbs, wie er insbesondere im Sport üblich ist.

Aus diesem Bedürfnis heraus entstand die Idee zu diesem Projekt. Die Grundidee stammt von Patrick Senn, Sportlehrer am Gymnasium Neufeld, der solche sportlichen Herausforderungen umsetzen wollte, aber an der fehlenden geeigneten Webplattform scheiterte. Dies führte zur Entwicklung einer neuen Webanwendung, die nicht nur sportliche Aktivitäten fördert, sondern auch technische Kompetenzen vermittelt.

Ein zentrales Anliegen meines Projekts ist es, Schülern aufzuzeigen, wie eine Webplattform von Grund auf entwickelt wird. Dabei werden die einzelnen Schritte des Entwicklungsprozesses speziell hervorgehoben und verständlich erklärt, sodass Schüler den "Wie"-Aspekt nachvollziehen können. Dies bietet eine einmalige Gelegenheit, technisches Wissen praktisch zu vermitteln und gleichzeitig das Interesse an Informatik und Webentwicklung zu fördern.

Ein besonderer Schwerpunkt liegt dabei auf dem Datenbankdesign. Ein robustes und gut strukturiertes Datenbankdesign ist entscheidend für die Funktionalität und Skalierbarkeit der Webanwendung. In diesem Projekt wird detailliert aufgezeigt, wie Datenbanken entworfen werden, um Daten effizient zu speichern und zu verwalten, wobei die Erklärungen speziell auf Schüler zugeschnitten sind.

1.2 Zielsetzung und Motivation

Das Hauptziel dieses Projekts ist die Entwicklung einer Plattform, die speziell auf den schulischen Kontext zugeschnitten ist. Dabei verfolge ich zwei zentrale Ziele:

Erstens möchte ich demonstrieren, wie eine solche Webplattform von Grund auf aufgebaut wird, indem ich die einzelnen Entwicklungsschritte klar und verständlich für Schüler hervorhebe. Dadurch sollen Schüler die Möglichkeit erhalten, den Prozess der Webentwicklung aktiv nachzuvollziehen und ein tieferes Verständnis für die technischen Hintergründe zu entwickeln.

Zweitens soll die eigentliche Plattform entstehen, die sportliche Aktivitäten fördert und nahtlos in den Schulalltag integriert wird. Der Fokus liegt dabei weniger auf dem äußeren Erscheinungsbild und mehr auf der Funktionalität. Durch die Betonung von Herausforderung und Wettbewerb möchte ich die Motivation der Teilnehmer steigern und langfristig eine tiefere Einbindung in den Schulkontext erreichen. So soll die Freude am sportlichen Engagement auf gymnasialer Stufe verstärkt werden.

Meine Motivation für dieses Projekt speist sich aus der Begeisterung für Sport und Wettbewerbe, die nicht nur die körperliche Fitness fördern, sondern auch den Teamgeist stärken. Gleichzeitig bietet das Projekt eine hervorragende Gelegenheit, praktische Kenntnisse in der Webentwicklung zu vermitteln. Besonders spannend ist es, Schülern zu zeigen, wie eine Webanwendung von Grund auf entwickelt wird—von der Planung über das Datenbankdesign bis hin zur Umsetzung—und dabei die technischen Aspekte verständlich zu erklären.

Ein weiterer Motivationsfaktor ist die intensive Beschäftigung mit der technischen Gestaltung eines robusten Datenbankdesigns. Da eine gut durchdachte Datenbankstruktur das Herzstück jeder erfolgreichen Webanwendung bildet, möchte ich diesen Bereich besonders hervorheben und Schülern die Grundlagen des Datenbankentwurfs näherbringen. Mit diesem Projekt will ich nicht nur sportliche Herausforderungen zugänglicher machen, sondern auch eine technische Lösung präsentieren, die als Lernplattform für weitere ähnliche Projekte dienen kann.

Insgesamt verbindet das Projekt meine Leidenschaft für Sport mit meinem Interesse an Informatik und der Freude daran, praktische Lösungen zu entwickeln. Durch die spezielle Hervorhebung der technischen Erklärungen möchte ich sowohl einen pädagogischen als auch sportlichen Mehrwert für die Schüler bieten.Projektplanung

2.1Projektumfang

**1.3 Projektplanung und Umfang**

Die Entwicklung einer Webanwendung erfordert eine sorgfältige Planung und eine klare Definition des Projektumfangs. Bei der Realisierung einer solchen Anwendung sind verschiedene Phasen und Aspekte zu berücksichtigen, um ein erfolgreiches Ergebnis zu gewährleisten. In diesem Abschnitt gebe ich einen Überblick darüber, **was bei der Planung und dem Umfang eines Webapplikationsprojekts alles dazugehört**, und erkläre, **wie ich diese Elemente in meinem Projekt "Einmal um die Schweiz" umsetzen werde**.

**1.3.1 Allgemeine Aspekte der Projektplanung**

Bei der Projektplanung einer Webanwendung sind folgende Schritte und Überlegungen essentiell:

* **Anforderungsanalyse**: Identifizierung der Bedürfnisse und Erwartungen der zukünftigen Nutzer sowie der Stakeholder. Dies umfasst funktionale Anforderungen (was die Anwendung tun soll) und nicht-funktionale Anforderungen (wie die Anwendung sein soll, z. B. in Bezug auf Performance oder Sicherheit).
* **Konzeption und Architektur**: Entwicklung eines Gesamtkonzepts für die Anwendung, inklusive der technischen Architektur. Hier werden Entscheidungen über die verwendeten Technologien, Frameworks und Plattformen getroffen.
* **Zeit- und Ressourcenplanung**: Festlegung des Projektzeitraums, Meilensteine und des benötigten Ressourcenbedarfs, sowohl personell als auch materiell.
* **Risikoanalyse**: Identifikation potenzieller Risiken und Herausforderungen, die den Projektverlauf beeinflussen könnten, sowie die Planung von Maßnahmen zur Risikominimierung.
* **Qualitätssicherung**: Definition von Standards und Methoden zur Sicherstellung der Qualität des Endprodukts, einschließlich Teststrategien und -verfahren.

**1.3.2 Allgemeiner Projektumfang einer Webanwendung**

Der Projektumfang definiert den Rahmen dessen, was das Projekt liefern wird. Dies beinhaltet:

* **Funktionalitäten**: Detaillierte Beschreibung der Features und Funktionen, die die Anwendung bieten wird.
* **Benutzeroberfläche und User Experience**: Gestaltung der Interaktion zwischen Nutzer und Anwendung, um eine intuitive und ansprechende Nutzung zu gewährleisten.
* **Datenmanagement**: Planung, **wie** Daten erfasst, gespeichert, verarbeitet und geschützt werden.
* **Sicherheit**: Maßnahmen zum Schutz der Anwendung und der Nutzerdaten vor unbefugtem Zugriff und Datenverlust.
* **Skalierbarkeit und Wartbarkeit**: Berücksichtigung der zukünftigen Erweiterbarkeit der Anwendung und der Einfachheit der Wartung.

2.3 Detaillierte Anforderungsanalyse

Anforderungsanalyse:

In meinem Projekt beginne ich mit einer gründlichen Sammlung und Analyse der Anforderungen. Durch direkte Gespräche und Interviews mit Lehrern und Schülern ermittle ich die spezifischen Bedürfnisse und Erwartungen an die Plattform. Dabei geht es nicht nur darum, welche Funktionen gewünscht sind, sondern auch darum, wie diese Funktionen die Nutzer unterstützen und den Schulalltag bereichern können.

Um die Anforderungen systematisch zu erfassen und zu dokumentieren, erstelle ich ein Pflichtenheft. Das Pflichtenheft ist ein zentrales Dokument in der Softwareentwicklung, das alle funktionalen und nicht-funktionalen Anforderungen detailliert festhält. Es dient als verbindliche Vereinbarung zwischen Auftraggeber und Entwickler und bildet die Grundlage für die weitere Projektarbeit. Den Schülern erkläre ich, wie ein Pflichtenheft aufgebaut ist und welche Elemente es enthält, wie zum Beispiel Zielsetzungen, Produktfunktionen, Benutzerschnittstellen und Qualitätsanforderungen.

Pflichtenheft für das Projekt "Einmal um die Schweiz"

Hinweis: Da es für dieses Projekt keinen offiziellen Auftraggeber gibt, fungiere ich selbst als Initiator und Entwickler. Dieses Pflichtenheft dient dazu, die Anforderungen und Ziele des Projekts klar zu definieren und als Leitfaden für die Umsetzung zu dienen.

1. Einleitung

1.1 Projektübersicht

"Einmal um die Schweiz" ist eine Webplattform, die sportliche Herausforderungen innerhalb und außerhalb der Schule fördert. Die Plattform ermöglicht es Schülern und Lehrern, an verschiedenen sportlichen Challenges teilzunehmen, ihre Leistungen zu erfassen und sich in einem freundlichen Wettbewerb miteinander zu messen.

1.2 Ziele des Projekts

Entwicklung einer benutzerfreundlichen Webplattform zur Förderung sportlicher Aktivitäten im schulischen Kontext.

Vermittlung technischer Kompetenzen an Schüler durch transparente Darstellung des Entwicklungsprozesses.

Schaffung einer nachhaltigen Lösung, die langfristig in den Schulalltag integriert werden kann.

2. Ausgangssituation und Motivation

Es existiert derzeit keine Plattform, die spezifisch auf die Bedürfnisse von Schulen und Schülern zugeschnitten ist, um sportliche Herausforderungen zu organisieren und zu verfolgen. Die Motivation für dieses Projekt liegt sowohl in der Förderung von Sport und Gemeinschaft als auch in der praktischen Anwendung von Kenntnissen in der Webentwicklung.

3. Geltungsbereich

Die Plattform richtet sich primär an:

Schüler und Lehrer des Gymnasiums Neufeld.

Potenziell auch an andere Bildungseinrichtungen und Sportgruppen mit ähnlichen Bedürfnissen.

4. Funktionale Anforderungen

4.1 Benutzerverwaltung

Registrierung und Anmeldung

Benutzer können sich registrieren und ein persönliches Profil erstellen.

Sicherer Login-Prozess mit Passwortverschlüsselung.

Rollenmanagement

Unterscheidung zwischen Schülern, Lehrern und Administratoren.

Unterschiedliche Berechtigungen je nach Rolle.

4.2 Challenge-Management

Erstellen von Challenges

Berechtigte Benutzer können neue Challenges erstellen.

Definition von Challenge-Details wie Titel, Beschreibung, Zeitraum und Zielsetzung.

Verwalten von Challenges

Bearbeiten und Löschen bestehender Challenges.

Verwaltung der Teilnehmer.

4.3 Teilnahme und Leistungserfassung

Teilnahme an Challenges

Benutzer können sich für Challenges anmelden.

Erfassung von Aktivitäten

Benutzer können ihre sportlichen Aktivitäten (z. B. gelaufene Kilometer) eintragen.

Möglichkeit, Aktivitäten zu bearbeiten oder zu löschen.

4.4 Auswertung und Ranglisten

Statistiken

Persönliche Leistungsübersicht für jeden Benutzer.

Ranglisten

Anzeige von Rankings nach individuellen und gruppenbezogenen Kriterien.

4.5 Benachrichtigungssystem

E-Mail-Benachrichtigungen

Bestätigung bei Registrierung und Teilnahme an Challenges.

In-App-Benachrichtigungen

Hinweise auf neue Challenges und Aktivitäten.

5. Nicht-funktionale Anforderungen

5.1 Benutzerfreundlichkeit

Intuitive Navigation und klare Benutzeroberfläche.

Responsives Design für verschiedene Endgeräte.

5.2 Sicherheit

Schutz personenbezogener Daten durch SSL-Verschlüsselung.

Sichere Speicherung von Passwörtern (Hashing, Salting).

5.3 Performance

Schnelle Ladezeiten und effiziente Datenbankabfragen.

Skalierbarkeit für steigende Nutzerzahlen.

5.4 Wartbarkeit

Gut strukturierter und dokumentierter Code.

Verwendung von gängigen Frameworks und Standards.

6. Technische Anforderungen

6.1 Frontend

HTML5, CSS3 und JavaScript.

Nutzung eines Frontend-Frameworks (z. B. React oder Vue.js).

6.2 Backend

Serverseitige Sprache: Python (Django) oder Node.js (Express).

RESTful API für die Kommunikation zwischen Frontend und Backend.

6.3 Datenbank

Relationale Datenbank (MySQL oder PostgreSQL).

Datenbankdesign gemäß Normalisierungsregeln.

6.4 Hosting

Deployment auf einem Cloud-Dienst (z. B. Heroku, AWS).

Automatisierte Backups und Updates.

Use Cases

# Use Cases für das interkantonale Schulprojekt "Around Switzerland"

## Use Case 1: Sportlehrperson registriert sich auf der Plattform

\*\*Akteur:\*\* Sportlehrperson

\*\*Ziel:\*\* Die Sportlehrperson registriert sich, um Zugang zur Webplattform zu erhalten.

\*\*Vorbedingungen:\*\* Keine

\*\*Nachbedingungen:\*\* Die Sportlehrperson ist registriert und kann sich anmelden.

\*\*Schritte:\*\*

1. Die Sportlehrperson navigiert zur Registrierungsseite.

2. Die Sportlehrperson gibt die notwendigen Informationen ein (z.B. Name, Schule, E-Mail-Adresse).

3. Die Sportlehrperson bestätigt die Eingabe und sendet das Formular ab.

4. Das System prüft die eingegebenen Daten auf Vollständigkeit und Korrektheit.

5. Das System erstellt ein Benutzerkonto für die Sportlehrperson und sendet eine Bestätigungsmail.

6. Die Sportlehrperson klickt auf den Bestätigungslink in der E-Mail, um die Registrierung abzuschließen.

## Use Case 2: Die Sportlehrperson registriert eine Klasse

\*\*Akteur:\*\* Die Sportlehrperson33

\*\*Ziel:\*\* Eine neue Klasse für die Teilnahme am Projekt registrieren.

\*\*Vorbedingungen:\*\* Die Sportlehrperson hat sich erfolgreich auf der Webplattform angemeldet.

\*\*Nachbedingungen:\*\* Die Klasse ist registriert und kann an dem Projekt teilnehmen.

\*\*Schritte:\*\*

1. Die Sportlehrperson navigiert zur Seite "Klasse registrieren".

2. Die Sportlehrperson gibt die notwendigen Informationen ein (z.B. Klassenbezeichnung, Schule).

3. Die Sportlehrperson bestätigt die Eingabe und sendet das Formular ab.

4. Das System prüft die eingegebenen Daten auf Vollständigkeit und Korrektheit.

5. Das System speichert die Klasse in der Datenbank und zeigt eine Erfolgsmeldung an.

## Use Case 3: Die Sportlehrperson trägt Laufaktivitäten von Schülern ein

\*\*Akteur:\*\* Die Sportlehrperson

\*\*Ziel:\*\* Laufaktivitäten von Schülern erfassen, um den Fortschritt der Klasse im Projekt zu verfolgen.

\*\*Vorbedingungen:\*\* Die Sportlehrperson hat sich erfolgreich auf der Webplattform angemeldet und die Klasse ist bereits registriert.

\*\*Nachbedingungen:\*\* Die Laufaktivitäten der Schüler sind erfasst und im Fortschritt der Klasse berücksichtigt.

\*\*Schritte:\*\*

1. Die Sportlehrperson navigiert zur Seite "Laufaktivitäten eintragen".

2. Die Sportlehrperson wählt die betreffende Klasse und den Schüler aus.

3. Die Sportlehrperson gibt die Distanz und das Datum der Laufaktivität ein.

4. Die Sportlehrperson bestätigt die Eingabe und sendet das Formular ab.

5. Das System prüft die eingegebenen Daten auf Vollständigkeit und Korrektheit.

6. Das System speichert die Laufaktivität in der Datenbank und zeigt eine Erfolgsmeldung an.

## Use Case 4: Klassen fordern sich gegenseitig heraus

\*\*Akteur:\*\* Die Sportlehrperson

\*\*Ziel:\*\* Eine Herausforderung zwischen zwei Klassen initiieren, um die Motivation und den Wettkampfgeist zu fördern.

\*\*Vorbedingungen:\*\* Die Sportlehrperson hat sich erfolgreich auf der Webplattform angemeldet und beide beteiligten Klassen sind registriert.

\*\*Nachbedingungen:\*\* Die Herausforderung ist eingeleitet und die Sportlehrperson der herausgeforderten Klasse erhält eine Benachrichtigung.

\*\*Schritte:\*\*

1. Die Sportlehrperson navigiert zur Seite "Herausforderung erstellen".

2. Die Sportlehrperson wählt die eigene Klasse als Herausforderer und die gegnerische Klasse als Herausgeforderte aus.

3. Die Sportlehrperson legt das Start- und Enddatum für die Herausforderung fest.

4. Die Sportlehrperson bestätigt die Eingabe und sendet das Formular ab.

5. Das System prüft die eingegebenen Daten auf Vollständigkeit und Korrektheit.

6. Das System speichert die Herausforderung in der Datenbank und sendet eine Benachrichtigung an die Sportlehrperson der herausgeforderten Klasse.

## Use Case 5: Die Sportlehrperson überprüft den Fortschritt der Klasse

\*\*Akteur:\*\* Sportlehrperson

\*\*Ziel:\*\* Die Sportlehrperson möchte den Fortschritt seiner Klasse in den verschiedenen Challenges überwachen.

\*\*Vorbedingungen:\*\* Die Sportlehrperson hat sich erfolgreich auf der Webplattform angemeldet und die Klasse hat an Challenges teilgenommen.

\*\*Nachbedingungen:\*\* Die Sportlehrperson erhält einen Überblick über den Fortschritt und die Leistungen der Klasse.

\*\*Schritte:\*\*

1. Die Sportlehrperson navigiert zur Seite "Klassenfortschritt".

2. Die Sportlehrperson wählt die Klasse aus, deren Fortschritt er überprüfen möchte.

3. Das System zeigt eine Übersicht der Teilnahme und Leistungen der Schüler in den verschiedenen Challenges an.

4. Die Sportlehrperson kann die Daten nach verschiedenen Kriterien (z.B. Challenge, Datum) filtern und sortieren.

5. Die Sportlehrperson kann die Übersicht als Bericht exportieren oder drucken.

8. Qualitätssicherung

Testverfahren:

Unit-Tests für einzelne Komponenten.

Integrationstests für das Zusammenspiel verschiedener Module.

Benutzerakzeptanztests mit einer Testgruppe von Schülern und Lehrern.

Code Review:

Regelmäßige Überprüfung des Codes auf Qualität und Effizienz.

Versionskontrolle:

Nutzung von Git für die Nachverfolgung von Änderungen.

9. Projektrisiken und Annahmen

Risiken:

Zeitliche Verzögerungen durch unerwartete technische Probleme.

Sicherheitsrisiken bei der Datenübertragung und -speicherung.

Annahmen:

Benutzer verfügen über Grundkenntnisse im Umgang mit Webanwendungen.

Internetzugang und moderne Browser sind verfügbar.

10. Zeitplan und Meilensteine

Phase 1 (2 Wochen): Anforderungsanalyse und Erstellung des Pflichtenhefts.

Phase 2 (3 Wochen): Design der Benutzeroberfläche und Datenbankmodellierung.

Phase 3 (6 Wochen): Implementierung von Frontend und Backend.

Phase 4 (3 Wochen): Testen und Fehlerbehebung.

Phase 5 (2 Wochen): Deployment und Einführung der Plattform.

Phase 6 (laufend): Wartung und Weiterentwicklung basierend auf Nutzerfeedback.

11. Dokumentation

Entwicklungsdokumentation:

Technische Spezifikationen.

Architekturdiagramme.

Benutzerdokumentation:

Benutzerhandbuch.

FAQs und Hilfeseiten.

Schulungsmaterialien:

Präsentationen für Workshops mit Schülern und Lehrern.

12. Schlussbemerkung

Dieses Pflichtenheft legt die Grundlage für die Entwicklung der Webplattform "Einmal um die Schweiz". Obwohl es keinen externen Auftraggeber gibt, ist es wichtig, die Anforderungen klar zu definieren, um ein zielgerichtetes und erfolgreiches Projekt zu gewährleisten. Durch die Einbindung von Use Cases und die Erstellung eines Pflichtenhefts möchte ich den Schülern zeigen, wie professionelle Softwareentwicklung geplant und umgesetzt wird.

2.4 Datenbankdesign

## 2. Datenbankdesign

Ein zentrales Element bei der Entwicklung einer Webanwendung ist das \*\*Datenbankdesign\*\*. Die Datenbank bildet das Rückgrat der Anwendung, da sie alle notwendigen Informationen speichert und verwaltet. Ein gut durchdachtes Datenbankdesign gewährleistet nicht nur die effiziente Speicherung und Abfrage von Daten, sondern auch die Skalierbarkeit und Wartbarkeit der Anwendung. In diesem Abschnitt werde ich detailliert erläutern, \*\*wie\*\* ich das Datenbankdesign für das Projekt "Einmal um die Schweiz" entwickelt habe, und dabei die einzelnen Schritte und Überlegungen erklären. Ziel ist es, den Schülern einen praxisnahen Einblick in den Prozess des Datenbankentwurfs zu geben.

### 2.1 Anforderungen an die Datenbank

Bevor mit dem eigentlichen Design begonnen wird, ist es wichtig, die Anforderungen an die Datenbank zu definieren. Dies umfasst die Identifikation aller Daten, die gespeichert werden müssen, sowie deren Beziehungen zueinander. Für das Projekt ergeben sich folgende Hauptanforderungen:

- \*\*Benutzerverwaltung\*\*: Speicherung von Benutzerinformationen wie Namen, E-Mail-Adressen, Passwörtern (verschlüsselt), Rollen (Schüler, Lehrer, Administrator) und Profilbildern.

- \*\*Challenges\*\*: Speicherung von Informationen zu den sportlichen Herausforderungen, einschließlich Titel, Beschreibung, Start- und Enddatum, Ersteller und Teilnehmer.

- \*\*Teilnahmen\*\*: Erfassung der Beziehung zwischen Benutzern und Challenges, d. h., welche Benutzer an welchen Challenges teilnehmen.

- \*\*Aktivitäten\*\*: Speicherung der von den Benutzern erfassten sportlichen Aktivitäten, z. B. Datum, Art der Aktivität, zurückgelegte Distanz oder Dauer.

- \*\*Ranglisten und Statistiken\*\*: Berechnung und Speicherung von Punkten, Ranglistenpositionen und anderen statistischen Daten.

### 2.2 Konzeptuelles Datenmodell (ER-Modell)

Nachdem die Anforderungen definiert sind, erstelle ich ein \*\*Entity-Relationship-Diagramm (ER-Diagramm)\*\*, um die Datenstruktur und die Beziehungen zwischen den verschiedenen Entitäten (Tabellen) zu visualisieren. Das ER-Modell dient als Grundlage für das logische und physische Datenbankdesign.

\*\*Hauptentitäten und Attribute:\*\*

1. \*\*Benutzer\*\*

- BenutzerID (Primärschlüssel)

- Vorname

- Nachname

- E-Mail (eindeutig)

- Passwort (verschlüsselt)

- Rolle (Schüler, Lehrer, Administrator)

- Registrierungsdatum

- Profilbild

2. \*\*Challenge\*\*

- ChallengeID (Primärschlüssel)

- Titel

- Beschreibung

- Startdatum

- Enddatum

- ErstellerID (Fremdschlüssel zu Benutzer)

- Erstellungsdatum

3. \*\*Teilnahme\*\*

- TeilnahmeID (Primärschlüssel)

- BenutzerID (Fremdschlüssel zu Benutzer)

- ChallengeID (Fremdschlüssel zu Challenge)

- Anmeldedatum

4. \*\*Aktivität\*\*

- AktivitätID (Primärschlüssel)

- TeilnahmeID (Fremdschlüssel zu Teilnahme)

- Datum

- Aktivitätstyp (z. B. Laufen, Radfahren)

- Wert (z. B. zurückgelegte Distanz)

- Einheit (z. B. Kilometer)

5. \*\*Punkte\*\*

- PunkteID (Primärschlüssel)

- TeilnahmeID (Fremdschlüssel zu Teilnahme)

- Gesamtpunkte

- Aktualisierungsdatum

\*\*Beziehungen zwischen den Entitäten:\*\*

- Ein \*\*Benutzer\*\* kann mehrere \*\*Challenges\*\* erstellen (1:n Beziehung zwischen Benutzer und Challenge).

- Ein \*\*Benutzer\*\* kann an mehreren \*\*Challenges\*\* teilnehmen, und eine \*\*Challenge\*\* kann von mehreren \*\*Benutzern\*\* absolviert werden (n:m Beziehung zwischen Benutzer und Challenge, realisiert durch die \*\*Teilnahme\*\*-Entität).

- Eine \*\*Teilnahme\*\* kann mehrere \*\*Aktivitäten\*\* haben (1:n Beziehung zwischen Teilnahme und Aktivität).

- Eine \*\*Teilnahme\*\* hat genau einen \*\*Punkte\*\*-Eintrag (1:1 Beziehung zwischen Teilnahme und Punkte).

### 2.3 Normalisierung der Datenbank

Die Normalisierung ist ein Prozess zur Organisation der Datenbankstrukturen, um Redundanzen zu minimieren und Dateninkonsistenzen zu vermeiden. In diesem Projekt habe ich die Datenbank bis zur \*\*dritten Normalform (3NF)\*\* normalisiert.

\*\*Erste Normalform (1NF):\*\*

- Sicherstellen, dass in jeder Tabelle die Attribute atomar sind, d. h., keine mehrfachen Werte in einem Attribut.

- Beispiel: Das Attribut "Vollname" wird in "Vorname" und "Nachname" aufgeteilt.

\*\*Zweite Normalform (2NF):\*\*

- Alle Nicht-Schlüsselattribute müssen vom gesamten Primärschlüssel abhängen.

- Da wir bei zusammengesetzten Primärschlüsseln arbeiten (z. B. in der \*\*Teilnahme\*\*-Tabelle), stellen wir sicher, dass alle Attribute vollständig vom Primärschlüssel abhängen.

\*\*Dritte Normalform (3NF):\*\*

- Alle Attribute sind nur vom Primärschlüssel abhängig und nicht von anderen Nicht-Schlüsselattributen.

- Beispiel: In der \*\*Aktivität\*\*-Tabelle gibt es keine Attribute, die von anderen Nicht-Schlüsselattributen abhängen.

### 2.4 Physisches Datenmodell und Datenbankschema

Nach der Erstellung des konzeptuellen Modells und der Normalisierung wird das physische Datenbankschema erstellt. Dies beinhaltet die Definition der Tabellen, Spalten, Datentypen, Schlüssel und Indexe in der gewählten Datenbankmanagementsystem (DBMS), z. B. MySQL oder PostgreSQL.

\*\*Beispiel für das Datenbankschema:\*\*

```sql

-- Tabelle für Benutzer

CREATE TABLE Benutzer (

BenutzerID SERIAL PRIMARY KEY,

Vorname VARCHAR(50) NOT NULL,

Nachname VARCHAR(50) NOT NULL,

Email VARCHAR(100) NOT NULL UNIQUE,

Passwort VARCHAR(255) NOT NULL,

Rolle VARCHAR(20) NOT NULL,

Registrierungsdatum TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

Profilbild VARCHAR(255)

);

-- Tabelle für Challenges

CREATE TABLE Challenge (

ChallengeID SERIAL PRIMARY KEY,

Titel VARCHAR(100) NOT NULL,

Beschreibung TEXT,

Startdatum DATE NOT NULL,

Enddatum DATE NOT NULL,

ErstellerID INT REFERENCES Benutzer(BenutzerID),

Erstellungsdatum TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Tabelle für Teilnahme

CREATE TABLE Teilnahme (

TeilnahmeID SERIAL PRIMARY KEY,

BenutzerID INT REFERENCES Benutzer(BenutzerID),

ChallengeID INT REFERENCES Challenge(ChallengeID),

Anmeldedatum TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE (BenutzerID, ChallengeID)

);

-- Tabelle für Aktivitäten

CREATE TABLE Aktivitaet (

AktivitaetID SERIAL PRIMARY KEY,

TeilnahmeID INT REFERENCES Teilnahme(TeilnahmeID),

Datum DATE NOT NULL,

Aktivitaetstyp VARCHAR(50) NOT NULL,

Wert DECIMAL(10,2) NOT NULL,

Einheit VARCHAR(20) NOT NULL

);

-- Tabelle für Punkte

CREATE TABLE Punkte (

PunkteID SERIAL PRIMARY KEY,

TeilnahmeID INT REFERENCES Teilnahme(TeilnahmeID),

Gesamtpunkte DECIMAL(10,2) NOT NULL,

Aktualisierungsdatum TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE (TeilnahmeID)

);

```

\*\*Erklärungen zu den Datenbankbefehlen:\*\*

- \*\*SERIAL PRIMARY KEY\*\*: Definiert eine automatisch inkrementierende Primärschlüsselspalte.

- \*\*VARCHAR(n)\*\*: Zeichenkette mit maximaler Länge \*n\*.

- \*\*REFERENCES\*\*: Definiert einen Fremdschlüssel zu einer anderen Tabelle.

- \*\*TIMESTAMP DEFAULT CURRENT\_TIMESTAMP\*\*: Speichert das aktuelle Datum und die Uhrzeit, wenn ein neuer Datensatz erstellt wird.

- \*\*UNIQUE\*\*: Erzwingt die Eindeutigkeit der angegebenen Spalte(n).

### 2.5 Umsetzung der Beziehungen und Constraints

Die Implementierung von Beziehungen und Constraints (Einschränkungen) ist entscheidend für die Datenintegrität. Durch die Verwendung von Fremdschlüsseln werden referenzielle Integrität gewährleistet, d. h., dass Verweise zwischen Tabellen gültig sind.

\*\*Beispiele:\*\*

- In der \*\*Teilnahme\*\*-Tabelle wird sichergestellt, dass ein Benutzer nicht mehrfach an derselben Challenge teilnehmen kann, durch die \*\*UNIQUE (BenutzerID, ChallengeID)\*\*-Constraint.

- Die \*\*Aktivität\*\*-Tabelle referenziert die \*\*Teilnahme\*\*-Tabelle, um zu wissen, zu welcher Teilnahme eine Aktivität gehört.

- Durch die \*\*ON DELETE CASCADE\*\*-Option kann festgelegt werden, dass beim Löschen eines Benutzers alle zugehörigen Teilnahmen, Aktivitäten und Punkte ebenfalls gelöscht werden.

\*\*Beispiel mit ON DELETE CASCADE:\*\*

```sql

-- Anpassung der Teilnahme-Tabelle

CREATE TABLE Teilnahme (

TeilnahmeID SERIAL PRIMARY KEY,

BenutzerID INT REFERENCES Benutzer(BenutzerID) ON DELETE CASCADE,

ChallengeID INT REFERENCES Challenge(ChallengeID) ON DELETE CASCADE,

Anmeldedatum TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UNIQUE (BenutzerID, ChallengeID)

);

```

### 2.6 Indizierung und Performance-Optimierung

Um die Abfragegeschwindigkeit der Datenbank zu erhöhen, setze ich Indexe auf häufig verwendete Spalten.

\*\*Beispiele für Indexe:\*\*

- \*\*E-Mail-Adresse\*\* in der \*\*Benutzer\*\*-Tabelle für schnelle Login-Abfragen.

- \*\*ChallengeID\*\* in der \*\*Teilnahme\*\*-Tabelle, um schnell alle Teilnehmer einer Challenge zu finden.

- \*\*TeilnahmeID\*\* in der \*\*Aktivität\*\*-Tabelle für schnelle Zugriffe auf die Aktivitäten eines Teilnehmers.

```sql

CREATE INDEX idx\_benutzer\_email ON Benutzer (Email);

CREATE INDEX idx\_teilnahme\_challenge ON Teilnahme (ChallengeID);

CREATE INDEX idx\_aktivitaet\_teilnahme ON Aktivitaet (TeilnahmeID);

```

### 2.7 Sicherheit und Datenschutz

Der Schutz der Benutzerdaten ist von größter Bedeutung. Hier einige Maßnahmen, die ich implementiere:

- \*\*Passwortverschlüsselung\*\*: Verwendung von sicheren Hashing-Algorithmen wie bcrypt zur Speicherung von Passwörtern.

- \*\*Zugriffsrechte\*\*: Einschränkung der Datenbankzugriffe durch definierte Benutzerrollen (z. B. Leser, Schreiber).

- \*\*Prepared Statements\*\*: Vermeidung von SQL-Injections durch die Verwendung vorbereiteter Anweisungen in der Anwendungsschicht.

- \*\*SSL/TLS-Verschlüsselung\*\*: Sicherstellung der verschlüsselten Datenübertragung zwischen Anwendung und Datenbankserver.

### 2.8 Backup und Wiederherstellung

Um Datenverlust zu vermeiden, plane ich regelmäßige Backups der Datenbank:

- \*\*Automatisierte tägliche Backups\*\*: Skripte, die die Datenbank in regelmäßigen Abständen sichern.

- \*\*Offsite-Backups\*\*: Speicherung der Backups an einem externen Ort.

- \*\*Wiederherstellungsplan\*\*: Dokumentation der Schritte zur Wiederherstellung der Datenbank im Notfall.

### 2.9 Dokumentation des Datenbankdesigns

Eine ausführliche Dokumentation erleichtert die Wartung und Weiterentwicklung der Datenbank:

- \*\*Datenbankdiagramme\*\*: Visuelle Darstellung der Tabellen und ihrer Beziehungen.

- \*\*Beschreibung der Tabellen und Spalten\*\*: Zweck und Datentypen erläutern.

- \*\*Kommentar im Code\*\*: Nutzung von Kommentaren in den SQL-Skripten zur Erklärung komplexer Strukturen.

\*\*Beispiel für eine Tabellenbeschreibung:\*\*

- \*\*Tabelle:\*\* Benutzer

- \*\*Zweck:\*\* Speicherung der grundlegenden Informationen aller Benutzer der Plattform.

- \*\*Wichtige Felder:\*\*

- \*\*Email:\*\* Eindeutige E-Mail-Adresse zur Identifizierung und Anmeldung.

- \*\*Passwort:\*\* Verschlüsseltes Passwort zur Authentifizierung.

- \*\*Rolle:\*\* Bestimmt die Berechtigungen innerhalb der Plattform.

### 2.10 Zusammenfassung und Ausblick

Das Datenbankdesign ist ein wesentlicher Bestandteil des Projekts "Einmal um die Schweiz". Durch die sorgfältige Planung und Umsetzung der Datenbankstrukturen stelle ich sicher, dass die Anwendung effizient, sicher und skalierbar ist. Die gewählten Entitäten und Beziehungen ermöglichen eine flexible Erweiterung der Plattform, beispielsweise durch zusätzliche Funktionen wie Gruppen-Challenges oder erweiterte Statistikfunktionen.

\*\*Für die Schüler bedeutet dieser Abschnitt:\*\*

- \*\*Verständnis des Datenbankentwurfsprozesses\*\*: Von der Anforderungsanalyse über das ER-Modell bis hin zur Implementierung.

- \*\*Anwendung von Normalisierungsprinzipien\*\*: Um eine effiziente und konsistente Datenstruktur zu gewährleisten.

- \*\*Kenntnis von Sicherheitsmaßnahmen\*\*: Zum Schutz sensibler Benutzerdaten.

- \*\*Bedeutung der Dokumentation\*\*: Für die Wartbarkeit und Verständlichkeit des Projekts.

In der weiteren Entwicklung der Plattform können die Schüler selbstständig Erweiterungen am Datenbankschema vornehmen und lernen, \*\*wie\*\* Änderungen sorgfältig geplant und implementiert werden müssen, um die Integrität der Daten zu bewahren.

### Anhang: Glossar wichtiger Begriffe

- \*\*Entität\*\*: Ein Objekt oder Konzept in der realen Welt, das in der Datenbank dargestellt wird (z. B. Benutzer, Challenge).

- \*\*Attribut\*\*: Eine Eigenschaft oder Merkmal einer Entität (z. B. Vorname, Nachname).

- \*\*Primärschlüssel\*\*: Ein Attribut oder eine Kombination von Attributen, die jeden Datensatz eindeutig identifiziert.

- \*\*Fremdschlüssel\*\*: Ein Attribut in einer Tabelle, das auf den Primärschlüssel einer anderen Tabelle verweist und so eine Beziehung zwischen den Tabellen herstellt.

- \*\*Normalisierung\*\*: Der Prozess der Organisation von Daten in einer Datenbank, um Redundanzen zu minimieren und die Datenintegrität zu gewährleisten.

- \*\*Index\*\*: Eine Datenstruktur, die den schnellen Zugriff auf Daten in einer Tabelle ermöglicht.

---

Mit diesem ausführlichen Datenbankdesign lege ich den Grundstein für eine stabile und effiziente Webanwendung. Durch die detaillierte Erläuterung der einzelnen Schritte und Entscheidungen möchte ich den Schülern vermitteln, \*\*wie\*\* wichtig ein durchdachtes Datenbankdesign für den Erfolg eines Softwareprojekts ist und \*\*wie\*\* man dabei methodisch vorgeht.

2.5 Technologiestack und Infrastruktur

3. Grundstruktur der Web-App

Ein Bild, das Text, Screenshot, Diagramm, Design enthält.

Automatisch generierte Beschreibung

Frontend

Backend

Datenbankanbindung

4.Integration und Tests

Testpläne und -verfahren

Ergebnisse der Tests

Feedback und Optimierung

5. Einholung von Feedback

Feedbackmethoden

Auswertung des Feedbacks

6.2 Iterative Verbesserungen

Identifizierte Probleme und Lösungen

Implementierte Verbesserungen

Deployment und Wartung

7.1 Deployment-Prozess

Vorbereitung und Durchführung

Rollout-Plan

7.2 Wartungsplan

Regelmäßige Wartungsarbeiten

Langfristige Wartungsstrategie

Fazit und Ausblick

8.1 Zusammenfassung der Ergebnisse

8.2 Herausforderungen und Lösungsansätze

8.3 Zukunftsperspektiven und Weiterentwicklun

Anhang

9.1 Glossar

Begriffe und Definitionen

9.2 Literaturverzeichnis

Quellen und Referenzen

9.3 Abbildungsverzeichnis

Übersicht der verwendeten Abbildungen

9.4 Tabellenverzeichnis

Übersicht der verwendeten Tabellen

9.5 Codebeispiele

Wichtige Codeausschnitte und Erklärungen