

Dokumentation Healix

1. Einleitung

Projekttitle und Zielsetzung

Das Ziel unserer App war es einen Fitness-Tracker zu erstellen in welchem verschiedenste Daten von Nutzern eingegeben, verarbeitet und in unserer App übersichtlich dargestellt werden können.

Mit der App wollten wir das Tracken der eigenen Fitness-Aktivität vereinfachen und Menschen auf ihrem Fitness-Weg unterstützen.

Kurzbeschreibung

Es folgt eine Kurzbeschreibung aller Hauptfunktionen von Healix.

Login/Register: Benutzer können sich einloggen oder registrieren. Bei der Registrierung werden Anmeldedaten sowie Benutzerdaten abgefragt, die für die Berechnung, Darstellung und Nutzung des Dashboards essenziell sind.

Dashboard: Das Dashboard bietet eine Übersicht über alle Tools, die zur Nutzung von Healix erforderlich sind.

Main: Die Main-Funktion zeigt eine Übersicht der Nutzerdaten, einschließlich allgemeiner Informationen wie BMI, Zielsetzungen (Goals) und Sportklasse (Sportclass). Zudem wird angezeigt, ob die gesetzten Ziele erreicht wurden.

Meal: Die Meal-Funktion bietet eine Übersicht der letzten Mahlzeiten und der aufgenommenen Kalorien. Basierend auf den Nutzerdaten werden Vorschläge für neue Mahlzeiten sowie weitere Statistiken angezeigt.

Fitness: Die Fitness-Funktion stellt die letzten Fitness-Einträge übersichtlich dar und bietet verschiedene Statistiken und Inhalte.

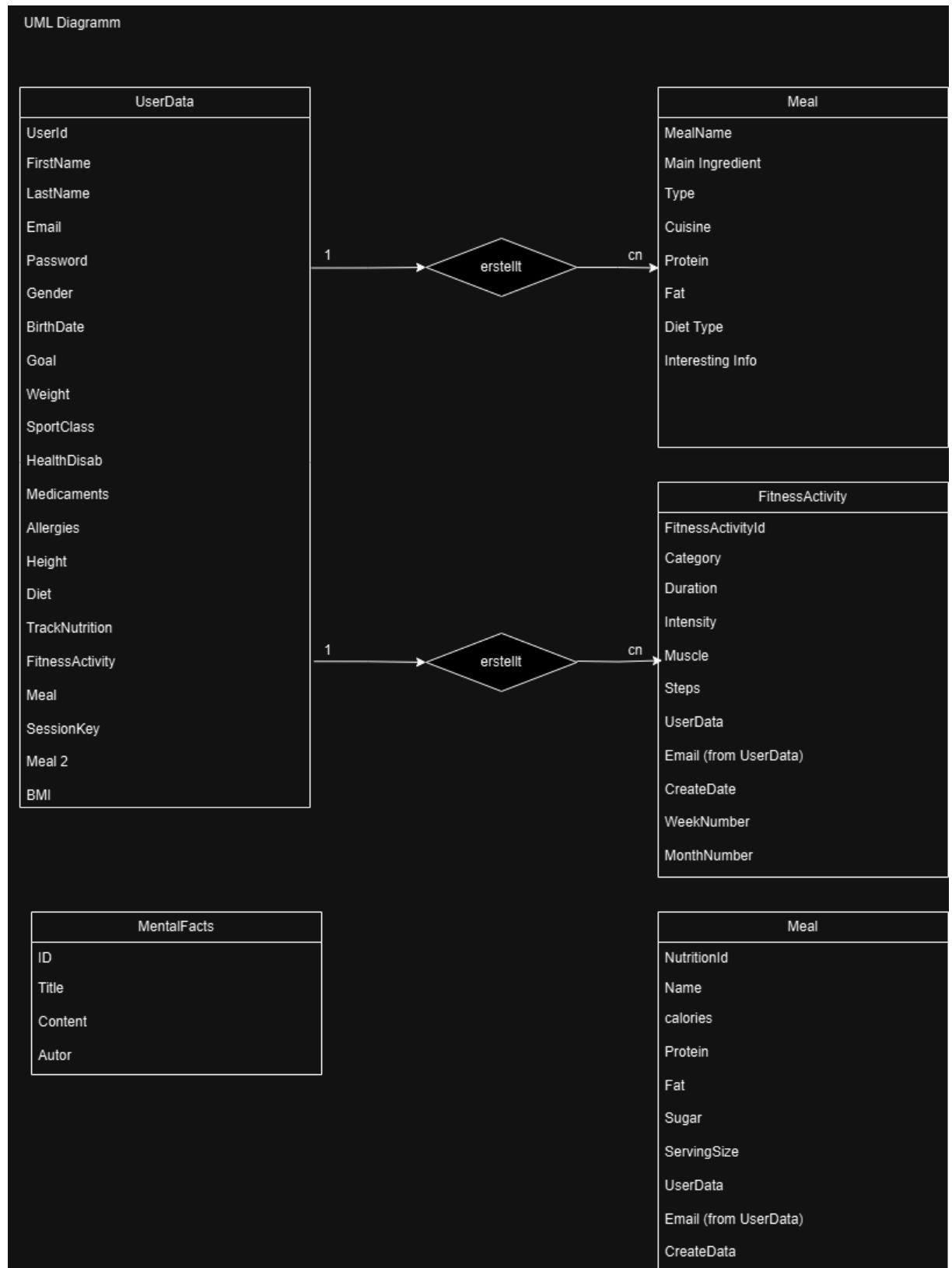
Activities: Die Activities-Funktion zeigt eine Übersicht der letzten Aktivitäten. Diese werden durch verschiedene UI-Elemente ansprechend und übersichtlich dargestellt.

New Meal: In dieser Funktion können Benutzer neue Mahlzeiten eintragen. Die Einträge werden gespeichert und können später in der Meals-Funktion eingesehen werden.

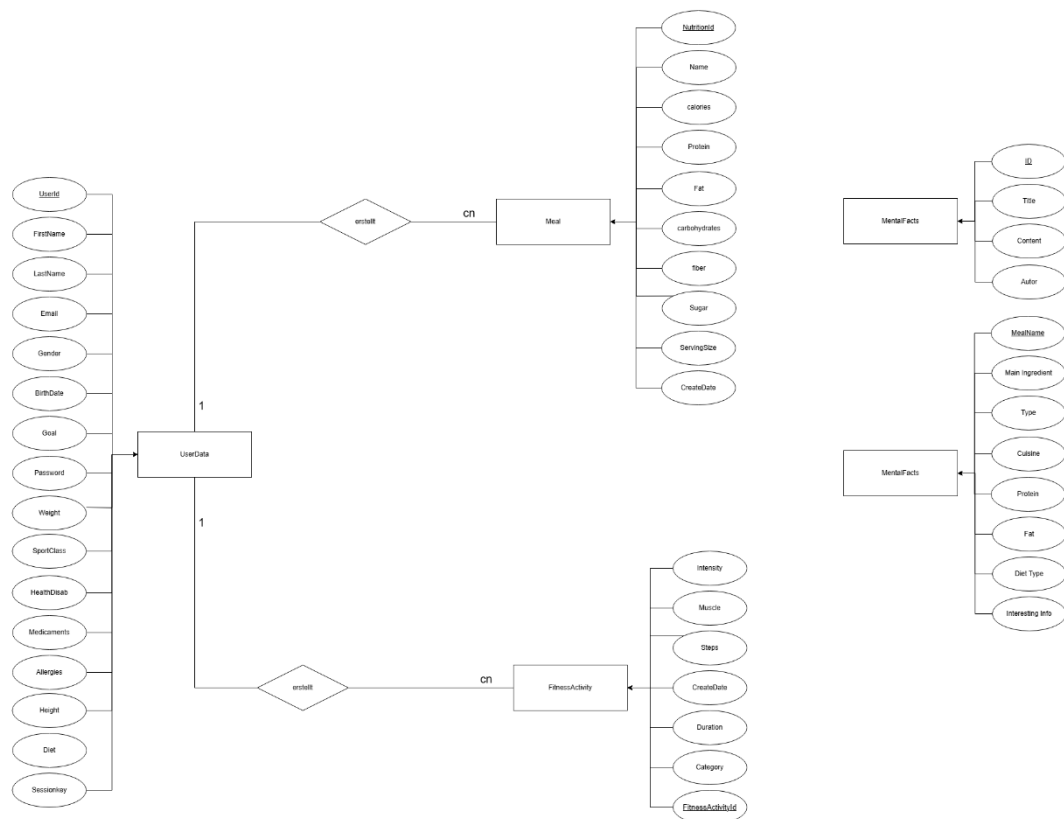
New Fitness Activity: Diese Funktion ermöglicht das Eintragen neuer Workouts. Die Einträge werden gespeichert und können später in der Fitness-Funktion angezeigt werden.

2. Konzept und Planung

Datenmodell



ER-Diagramm



Beschreibung der Relationen

Relationenbeschreibung:

1. UserData erstellt Meal

Beschreibung: Ein Benutzer kann mehrere Mahlzeiten erstellen, jede Mahlzeit gehört zu genau einem Benutzer.

Kardinalität: 1 (UserData) → 0..n (Meal).

2. UserData erfüllt FitnessActivity

Beschreibung: Ein Benutzer kann mehrere Fitnessaktivitäten durchführen, jede Fitnessaktivität ist einem Benutzer zugeordnet.

Kardinalität: 1 (UserData) → 0..n (FitnessActivity).

3. Meal enthält Nutrition

Beschreibung: Jede Mahlzeit hat genau eine zugehörige Nährwertangabe (Nutrition).

Kardinalität: 1 (Meal) → 1 (Nutrition).

4. MentalFacts (unabhängiges Objekt)

Beschreibung: MentalFacts sind allgemeine Tipps oder Fakten zu Mahlzeiten und Fitness, unabhängig von Benutzern oder anderen Objekten.

Funktionsumfang

Hauptfunktionen:

1. Register/Login

- Enter UserData: Persönliche Daten eingeben.
- Enter FitnessData: Fitnessrelevante Informationen hinzufügen.
- Enter NutritionData: Ernährungsdetails eintragen.

2. Dashboard

- Zentrale Übersicht über persönliche Statistiken und Aktivitäten.

3. New Activity

- Neue sportliche Aktivitäten hinzufügen und verwalten.

4. New Meal

- Mahlzeiten dokumentieren und analysieren.

Zusatzfunktionen:

1. Profile

- Nutzerprofil anpassen und personalisieren.

2. Tipps

- Individuelle Ratschläge zu Fitness und Ernährung anzeigen.

Priorisierung:

Must-Have:

- Register/Login (inkl. Enter UserData, Enter FitnessData, Enter NutritionData)
- Dashboard
- New Activity
- New Meal

Nice-to-Have:

- Profile
- Tipps

Vergleich:

Bis auf die Zusatzfunktionen wurden alle geplanten Funktionen implementiert. Durch die Entwicklung hat sich ergeben, dass die geplanten Zusatzfunktionen nicht effizient mit den Hauptfunktionen zusammenarbeiten würden. Sie wurden anders im Programm implementiert.

3. Umsetzung

Technologien und Tools

Vue.js: Vue.js wurde als Framework gewählt, da es durch die komponentenbasierte Architektur eine effiziente Entwicklung ermöglicht. Zudem besteht viel Erfahrung mit Vue, was die Umsetzung beschleunigt und Fehler minimiert.

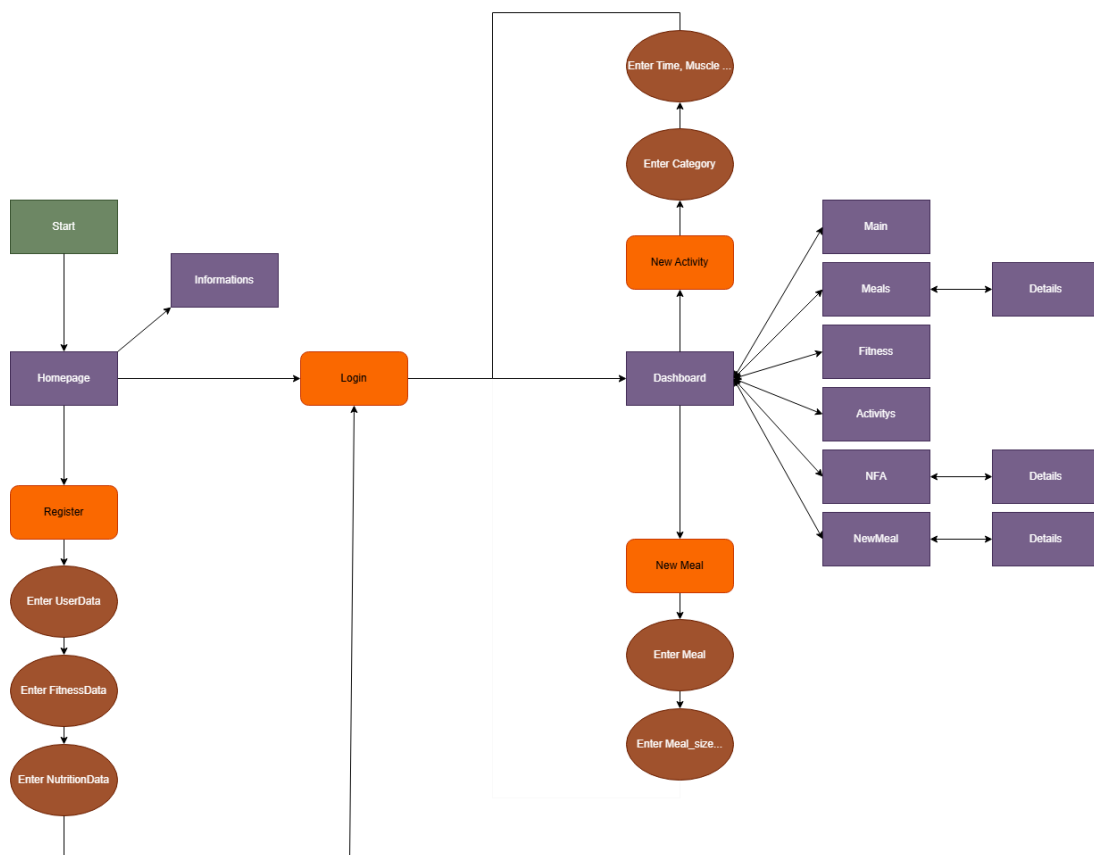
Chart.js: Chart.js wurde für die Datenvisualisierung genutzt, da es eine breite Auswahl an Diagrammtypen bietet und sich hervorragend in Vue.js integrieren lässt. Die bereits vorhandene Erfahrung mit Chart.js ermöglichte eine schnelle und zuverlässige Umsetzung.

CSS: CSS wurde verwendet, um das Layout und Design der Anwendung flexibel und individuell anzupassen. Es bietet vollständige Kontrolle über das Erscheinungsbild und stellt sicher, dass die Anwendung responsiv ist und auf verschiedenen Geräten gut funktioniert.

Airtable: Airtable wurde für die Datenverwaltung gewählt, da es einfach zu bedienen ist und sich gut für kleinere Projekte eignet. Es wurde zudem im Unterricht empfohlen, was die Entscheidung zusätzlich unterstützt hat.

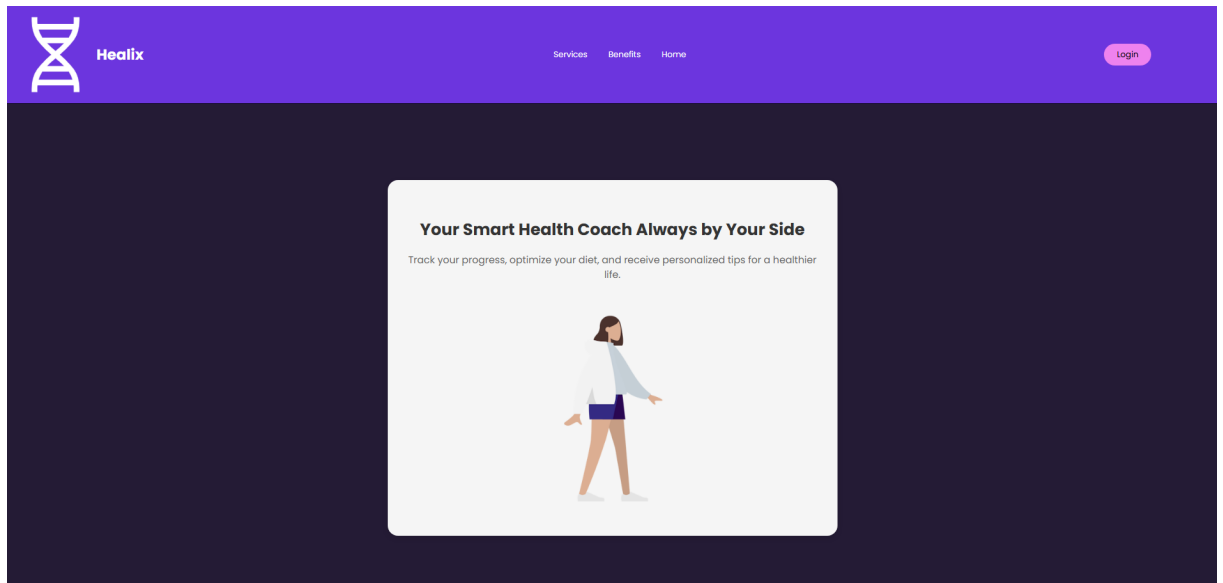
4. Ergebnisse

Funktionsumfang

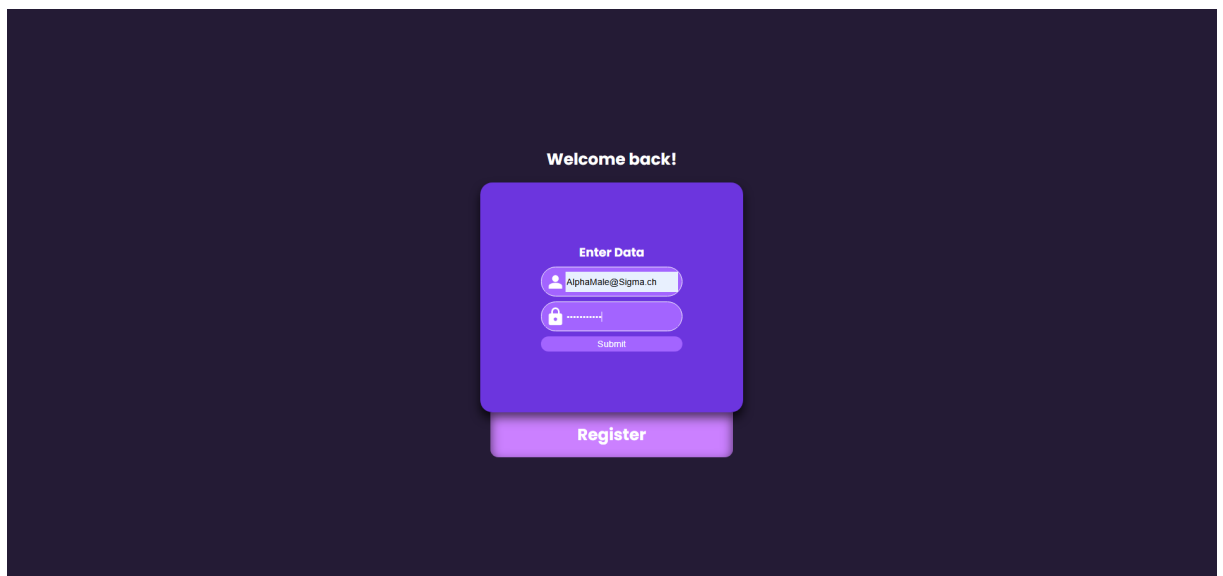


Benutzeroberfläche

Screenshots der fertigen UI



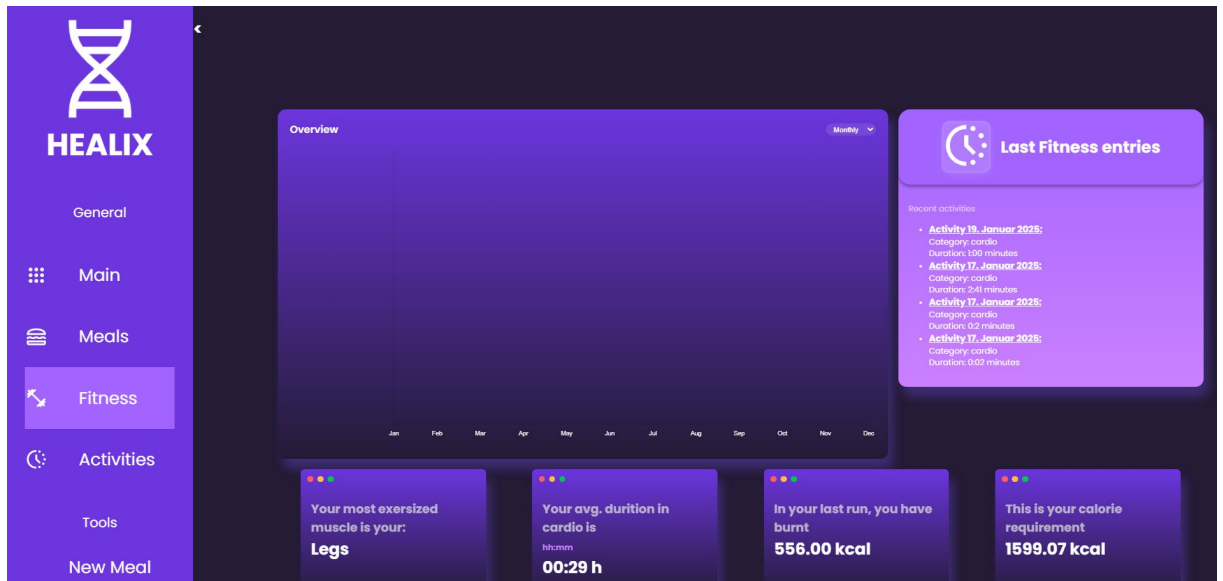
Home Seite



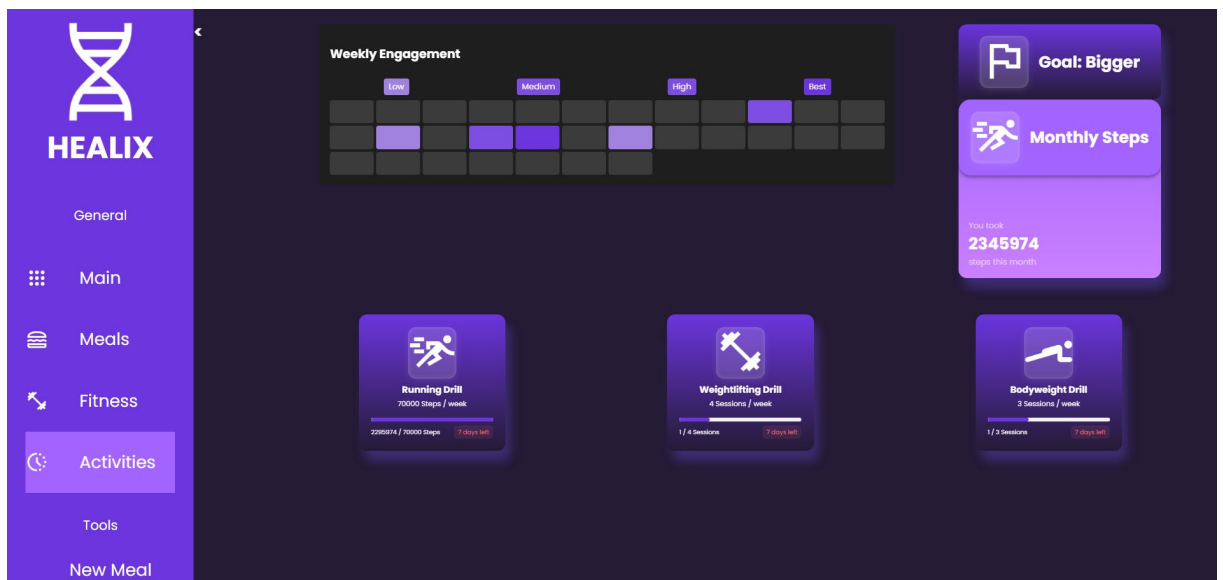
Login

Register





Dashboard/Fitness



Dashboard/Activities

The dashboard features a sidebar menu with the following items: General, Main, Meals, Fitness, Activities, Tools, and New Meal (selected). The main content area includes a 'Have you eaten something?' prompt and a 'New Meal' form.

Have you eaten something? Enter it precisely, and we'll calculate it for you!

Fill in the fields to get the most accurate results

Meals Information

Answer the form

Meal Name:

Calories:

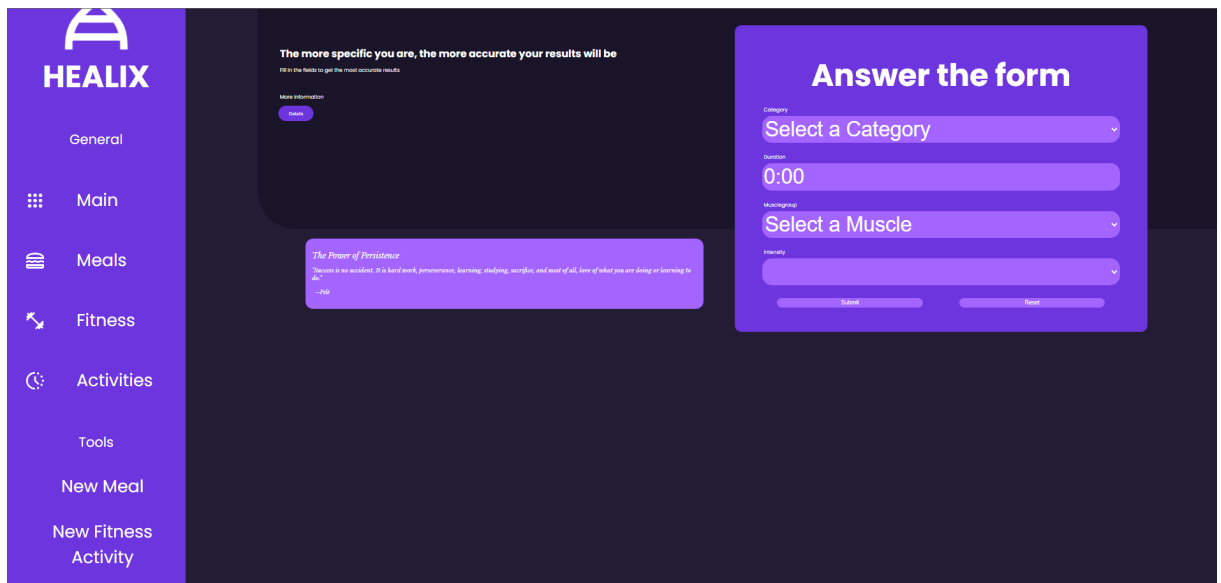
Protein:

Fat:

Sugar:

Serving Size:

Dashboard/New Meal



Dashboard/New Fitness Activity

User Manual

Im ReadMe enthalten

5. Herausforderungen und Lösungen

Probleme während der Entwicklung

Technische Herausforderungen

- **Airtable-Verbindung aufbauen:** Da dies das erste Mal war, dass wir mit Airtable gearbeitet haben, gab es einige Schwierigkeiten bei der Integration und dem Aufbau der Verbindung. Die Einarbeitung in die spezifischen Anforderungen und Funktionen von Airtable nahm mehr Zeit in Anspruch als ursprünglich geplant.

Teamarbeit

- **Planung im Voraus:** Die Planung des Projekts war aufgrund der flexiblen Arbeitsweise der Teammitglieder und der Notwendigkeit, auch außerhalb des Unterrichts zu arbeiten, eine Herausforderung. Es war schwierig, konkrete Zeiten für die Arbeit festzulegen, was zu Unklarheiten und Verzögerungen führte.
- **Arbeitsrapport:** Der Arbeitsrapport war mühsam zu nutzen. Das Layout und die Struktur des Berichts haben den Überblick erschwert und die Erfassung von Fortschritten und Aufgaben war weniger effizient. Für zukünftige Projekte wird ein anderes Layout für den Rapport empfohlen.
- **Code-Dokumentation:** Der Rapport für den Code war weniger geeignet, um eine klare und strukturierte Dokumentation zu gewährleisten. In Zukunft sollte der Code eher über Git verwaltet werden, um eine bessere Nachverfolgbarkeit und Zusammenarbeit zu ermöglichen.
- **Arbeitsverteilung:** Die Verteilung der Aufgaben konnte nicht immer optimal an die Stärken der beiden Teammitglieder angepasst werden. Um die Effizienz zu steigern,

sollte die Aufgabenzuteilung stärker an den individuellen Fähigkeiten und Stärken der Teammitglieder ausgerichtet werden.

Lösungsansätze

Lehrer fragen: Bei auftretenden Schwierigkeiten haben wir gezielt den Lehrer um Unterstützung gebeten. Dies half, spezifische technische oder organisatorische Probleme schnell zu klären und Lösungen zu finden.

Austausch in der Klasse: Ein weiterer Lösungsansatz war der Austausch mit anderen Schülern und Teammitgliedern in der Klasse. Durch die gemeinsame Diskussion konnten Ideen und Lösungen entwickelt werden, die individuell nicht sofort erkennbar gewesen wären.

Einsatz von AI und Internet: Bei Bedarf haben wir auch künstliche Intelligenz (AI) und das Internet genutzt, um schnelle Antworten und Lösungen zu finden. Online-Ressourcen und AI-gestützte Tools haben oft geholfen, technische und programmiertechnische Probleme zu lösen.

Keine ungelösten Probleme: Es gab keine Probleme, die nicht gelöst werden konnten. Dank der verschiedenen Hilfsmittel und der Zusammenarbeit im Team konnten alle Herausforderungen erfolgreich gemeistert werden.

6. Technische-Details

Setup-Anleitung

Alle Setup-Anleitungen sind im ReadMe enthalten.

7. Anhang

Quellcode

Link zu unserem git-repository: https://gitlab.com/LuisAll12/healix_ipt4.1

Einladungsmail wurde an michael.wicki@sluz.ch versandt.

Ressourcen

Alle benutzten Ressourcen wurden bereits in der Dokumentation benannt.