Prototyping Projektdokumentation

Name: Lisi Useini

E-Mail: useinlis@students.zhaw.ch

URL der deployten Anwendung: https://fittrackster.netlify.app/

# Einleitung

FitTrack ist eine einfache Webanwendung zur Verwaltung von Workouts und Übungen. Die App wurde im Rahmen des Moduls Prototyping mit SvelteKit, MongoDB, HTML/CSS (Bootstrap) und dem Runes Mode umgesetzt. Ziel der Anwendung ist es, NutzerInnen eine intuitive Möglichkeit zu bieten, ihre Trainingsroutinen digital zu erfassen, anzuzeigen, zu bearbeiten und bei Bedarf zu löschen.

Die App ist bewusst minimalistisch gehalten, um sich auf die Kernfunktionalitäten zu konzentrieren. Workouts mit Dauer und Beschreibung verwalten sowie spezifische Übungen mit Wiederholungen, Sätzen und Zielmuskeln zu jedem Workout hinzufügen. Die Daten werden über ein serverseitiges API mit MongoDB verarbeitet.

FitTrakc zeigt beispielhaft, wie moderne Webentwicklung mit einem komponentenbasierten Framework und einer echten Datenbankanbindung umgesetzt werden kann. Darüber hinaus wurden kleinere Erweiterungen implementiert, um die Anwendung abzurunden, darunter Live-Vorschauen und ein motivierendes Zitat-Widget via API.

# Datenmodell

Ein Bild, das Screenshot, Kreis enthält.

KI-generierte Inhalte können fehlerhaft sein.

Das Datenmodell zeigt eine 1:n-Beziehung zwischen Workout und Exercise.  
Ein Workout kann mehrere Exercises enthalten, aber jede Exercise gehört genau einem Workout.  
Zusätzlich werden die wichtigsten Attribute dargestellt.

# Beschreibung der Anwendung

Die Anwendung FitTrack ermöglicht es den NutzerInnen, eigene Workouts und Übungen zu erstellen, zu verwalten und zu visualisieren. Die App richtet sich an sportlich interessierte, die ihre Trainingsdaten übersichtlich erfassen möchten.

## Startseite

Route: /

# Ein Bild, das Text, Screenshot, Pferd, Website enthält. KI-generierte Inhalte können fehlerhaft sein.

Auf der Startseite wird die Anwendung vorgestellt. In der Mitte befindet sich ein Bild mit ästhetischem Bezug zum FitTrack-Thema, darunter zwei Buttons zum Aufrufen der Workouts oder zum Erstellen eines neuen Workouts.

Am unteren Rand der Startseite wird ein Motivationszitat angezeigt, das per API-Aufruf dynamisch geladen wird.

Dateien:

* lib/components/Navbar.svelte
* routes/+page.svelte
* routes/+page.server.js

## Workouts-Übersicht

Route: /workouts

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf der Startseite wird die Anwendung vorgestellt. In der Mitte befindet sich ein Bild mit ästhetischem Bezug zum FitTrack-Thema, darunter zwei Buttons zum Aufrufen der Workouts oder zum Erstellen eines neuen Workouts.

Am unteren Rand der Startseite wird ein Motivationszitat angezeigt, das per API-Aufruf dynamisch geladen wird.

Dateien:

* lib/components/Navbar.svelte
* lib/components/WorkoutCard.svelte
* routes/workouts/+page.svelte
* routes/workouts/+page.server.js

## Workout-Detailseite

Route: /workouts/[id]

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite werden die Details eines einzelnen Workouts angezeigt. Dazu gehören der Name, die Dauer und die Beschreibung.  
Zusätzlich wird eine Liste aller zugehörigen Übungen angezeigt. Jede Übung wird mit Name, Satz- und Wiederholungsanzahl sowie Zielmuskel dargestellt.  
  
Am Ende der Seite befindet sich ein Button «Zurück zur Übersicht», der zurück zur Workout-Übersicht führt.

Dateien:

* lib/components/Navbar.svelte
* routes/workouts/[id]/+page.svelte
* routes/workouts/[id]/+page.server.js

## Neues Workout erfassen

Route: /new-workout

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.

Diese Seite dient der Erstellung eines neuen Workouts. Über ein einfaches Formular können Name, Dauer (in Minuten) sowie eine Beschreibung eingegeben werden. Nach dem Klick auf «Speichern» wird das Workout gespeichert und in der Übersicht angezeigt.

Dateien:

* lib/components/Navbar.svelte
* routes/new-workout/+page.svelte
* routes/new-workout/+page.server.js

## Workout bearbeiten

Route: /workouts/[id]/edit

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

KI-generierte Inhalte können fehlerhaft sein.

Diese Seite ermöglicht es, eines bestehenden Workouts zu ändern. Die Felder Name, Dauer (Minuten) und Beschreibung sind bereits mit den aktuellen Werten vorausgefüllt. Nach dem Klick auf «Änderungen speichern» werden die Änderungen übernommen und gespeichert. Mit «Zurück» gelangt man zurück zur Übersicht.

Dateien:

* lib/components/Navbar.svelte
* routes/workouts/[id]/edit/+page.svelte
* routes/workouts/[id]/edit/+page.server.js

## Übungen-Übersicht

Route: /exercises

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite werden alle erfassten Übungen aufgelistet. Jede Übung wird mit ihrer Zielmuskelgruppe, den Wiederholungen und den Sätzen dargestellt. Die Darstellung erfolgt über die Komponente «ExerciseCard.svelte». Über die Buttons «Löschen» und «Bearbeiten» kann eine Übung gelöscht oder angepasst werden. Mit dem grünen Button «+ Neue Übung» gelangt man zur Eingabeseite für neue Übungen.

Dateien:

* lib/components/Navbar.svelte
* lib/components/ExerciseCard.svelte
* routes/exercises/+page.svelte
* routes/exercises/+page.server.js

## Neue Übung erfassen

Route: /new-exercise

Ein Bild, das Text, Screenshot, Design enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite kann eine neue Übung erfasst und einem bestehenden Workout zugewiesen werden. Das Formular enthält Eingabefelder für den Namen der Übung, die Zielmuskelgruppe, die Anzahl Wiederholungen und Sätze sowie ein Dropdown zur Auswahl eines Workouts. Nach dem Absenden wird die Übung gespeichert und auf der Übersichtsseite angezeigt.

Dateien:

* lib/components/Navbar.svelte
* routes/new-exercise/+page.svelte
* routes/new-exercise/+page.server.js

## Übung bearbeiten

Route: /exercises/[id]/edit

Ein Bild, das Text, Screenshot, Reihe, Quittung enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite kann eine bestehende Übung angepasst werden. Die Eingabefelder sind vorausgefüllt mit den bestehenden Werten der Übung. Nach dem Bearbeiten und Speichern wird die Übung aktualisiert und die NutzerInnen zurück zur Übersicht weitergeleitet.

Dateien:

* lib/components/Navbar.svelte
* routes/exercises/[id]/edit/+page.svelte
* routes/exercises/[id]/edit/+page.server.js

# Erweiterungen

Neben den grundlegenden Anforderungen wurden im Projekt mehrere sinnvolle Erweiterungen umgesetzt, die den Funktionsumfang und die Benutzerfreundlichkeit der Anwendung erhöhen. Im Folgenden werden diese Erweiterungen inklusive Verortung im Code beschrieben.

## Zitat-API (Motivationszitat)

Ein Bild, das Text, Screenshot, Software, Website enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf der Startseite wird am unteren Rand ein motivierendes Zitat angezeigt, das per API-Aufruf dynamisch geladen wird. Dazu wird bei jedem Seitenaufruf zufällig ein Zitat von der «DummyJSON Quotes API» geladen und im entsprechenden Bereich dargestellt. Diese Erweiterung erhöht den individuellen Touch und belebt die Startseite.

Dateien:

* routes/+page.svelte
* routes/+page.server.js

## Volumenvergleichs-Diagramm

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf der Seite /workouts wird oberhalb der Workouts ein Balkendiagramm zum Volumenvergleich angezeigt. Es zeigt das Trainingsvolumen pro Workout aggregiert. Dadurch können NutzerInnen schnell sehen, welche Workouts die höchste Trainingslast erzeugen.

Die Berechnung und Visualisierung erfolgt mit Chart.js, das clientseitig in der Datei routes/workouts/+page.svelte eingebunden ist. Die Daten werden aufbereitet und an das Diagramm übergeben.

Dateien:

* Lib/components/WorkoutVolumeChart.svelte
* routes/workouts/+page.svelte

## Live-Vorschau bei der Übungserstellung

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

KI-generierte Inhalte können fehlerhaft sein.

Beim Erstellen einer neuen Übung auf der seite /new-exercise wird live eine Vorschau der aktuellen Eingabedaten angezeigt. So können NutzerInnen in Echtzeit sehen, wie die Übungskarte später aussehen wird – inklusive Name, Muskelgruppe, Wiederholungen, Sätze und ausgewähltem Workout.

Dies wurde durch lokale Bindungen und eine separate Vorschaukomponente (ExerciseCard.svelte) realisiert.

Dateien:

* Lib/components/ExerciseCard.svelte
* routes/new-exercise/+page.svelte