Prototyping Projektdokumentation

Name: Loïc Kaufmann

E-Mail: kaufmloi@students.zhaw.ch

URL der deployten Anwendung: https://habittrackerln2.netlify.app

# Einleitung

App-Name: DailyHabit

DailyHabbit ist eine Webanwendung, welche zur Unterstützung beim Aufbau und Verfolgen von gesunden Gewohnheiten dienen sollen. Die App erlaubt es den Benutzern, eigene Habits zu definieren, diese zu tracken und auszuwerten. Die Grundidee der App ist es langfristig Verhaltensweisen durch kontinuierliche Wiederholung zu etablieren. Diese werden anhand Tagesfortschritten, Streaks und einer Monatsübersicht visualisiert.

Die Grundfunktionen der App sind:

* Erstellen, Anzeigen und Löschen von Habits
* Tracken der täglichen und wöchentlichen Habit-Erfüllungen
* Anzeige von Fortschritt
* Streak-Logik
* Kalender
* Kategorisierung von Habits

# Datenmodell

Ein Bild, das Diagramm, Kreis, Schrift, Entwurf enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Text, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

Die Verknüpfung zwischen habits und categories erfolgt aktuell nicht über einen Foreign Key, sondern über den Kategorie-Namen als String (habits.category). In jedem Habit wird die zugehörige Kategorie als Text gespeichert der mit dem Feld (name) in der Collection categories übereinstimmt. Dadurch bleibt das Datenmodell einfach, jedoch besteht keine direkte relationale Referenz zwischen den Collections.

# Beschreibung der Anwendung

## Intro-Seite

**Route: /**

**Ein Bild, das Text, Software, Screenshot, Webseite enthält.

KI-generierte Inhalte können fehlerhaft sein.**

Beim ersten Aufruf der Anwendung wird eine Intro Seite mit einem Demo Video angezeigt. Das Video stellt die Grundidee und die Funktionen von «DailyHabit» vor und enthält Hintergrundmusik. Durch das Klicken auf den darunterliegenden Button gelangt der Benutzer auf die Tagesübersicht /overview weiter.

Dateien:

* routes/+page.svelte
* static/DemoDailyHabit.mp4

## Tagesübersicht

**Route: /overview**

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

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite sieht der Benutzer alle für heute definierten Health Habits. Oben auf der Übersichtsseite wird eine Fortschrittsanzeige der erfüllten Habits angezeigt über die Komponente (ProgressBar.svelte). Es wird ebenfalls ein Überblick über den Tag (Komponente DailyOverview.svelte) angezeigt.

Darunter werden die Habits als Karten dargestellt durch die Komponente HabitCard.svelte, jeweils mit Button um diese als «Erledigt» zu markieren. Habits die bereits für den Tag erfüllt wurden, sind optisch markiert.

Dateien:

* lib/components/ProgressBar.svelte
* lib/components/DailyOverview.svelte
* lib/components/HabitCard.svelte
* routes/overview/+page.svelte
* routes/overview/+page.server.js

## Alle Habits

**Route: /habits**

**Ein Bild, das Text, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.**Hier werden alle angelegten Habits aufgelistet welche in der Datenbank gespeichert sind. Jeder Eintrag zeigt den dazugehörigen Titel, Beschreibung und die Kategorie. Per Button gelangt man auf die Detailseite des Habits.

Dateien:

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

## Neues Habit erstellen

**Route: /habits/new**

**Ein Bild, das Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.**

Hier kann der Benutzer ein neues Habit erfassen. Es gibt Eingabefelder für Name, Beschreibung und die Kategorie (Dropdown aus der Kategorie-Collection). Das Formular POST speichert die Daten anschliesend in MongoDB

Dateien:

* routes/habits/new/+page.svelte
* routes/habits/new/+page.server.js
* lib/db/habits.js
* lib/db/categories.js

## Habits-Details

**Route: /habits/[id]**

**Ein Bild, das Text, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.**

Diese Ansicht zeigt die Detailansicht eines Habits mit Beschreibung, Kategorie und Erstellungsdatum. Darunter wird der Verlauf als Liste der einzelnen Einträge angezeigt. Zusätzlich wird der aktuelle Streak (aufeinanderfolgende Erfüllungen) eingeblendet. Ein Habt kann von der Detailansicht auch gelöscht oder bearbeitet werden.

Dateien:

* routes/habits/[id]/+page.svelte
* routes/habits/[id]/+page.server.js

## Habit bearbeiten

**Route : /habits/[id]/edit**

**Ein Bild, das Text, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.**

Hier kann ein bestehendes Habit bearbeitet werden. Hier kann ebenfalls der Name, Beschreibung und eine Kategorie gewählt werden. Die Kategorie Auswahl erfolgt ebenfalls per Dropdown.

Dateien:

* routes/habits/[id]/edit/+page.svelte
* routes/habits/[id]/edit/+page.server.js

## Kategorien verwalten

**Route : /categories**

**Ein Bild, das Text, Zahl, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.**

Diese Seite zeigt alle vorhandenen Kategorien an. Neue Kategorien können hinzugefügt, bestehende gelöscht werden. Dies erfolgt direkt über einfache Formulareingaben. Die Kategorien werden dann anschliessend in einer separaten MongoDB Collection gespeichert.

Dateien:

* routes/categories/+page.svelte
* routes/categories/+page.server.js
* lib/db/categories.js

## Statistiken

**Route: /stats**

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

KI-generierte Inhalte können fehlerhaft sein.**

Diese Seite zeigt eine Auswertung über den Fortschritt und die Entwicklung der letzten Tage. Folgende Elemente sind enthalten:

* Anzahl der erfüllten Habits heute (wie auf Startseite)
* Verlauf der letzten 7 Tage
* Aktuelle Streaks aller Habits
* Monatskalender mit Tagesfortschritt pro Tag (Komponente CalendarView.svelte)

Dateien:

* routes/stats/+page.svelte
* routes/stats/+page.server.js
* lib/components/CalendarView.svelte

# Erweiterungen

## Streak-Logik:

Ein Bild, das Screenshot, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf der Statistiksteite /stats und der Detailseite /habits/[id] werden für jedes Habit aufeinanderfolgende erfüllte Tage einer Habit als «Streaks» gezählt und angezeigt werden. Die Berechnung der «Streak-Logik» erfolgt über die Funktion getStreakForHabit().

Dateien:

* lib/db/entries.js
* routes/stats/+page.server.js
* routes/habits/[id]/+page.server.js

## Kalenderansicht:

Ein Bild, das Text, Quittung enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf der Statistikseite /stats wird ein Monatskalender angezeigt, der den täglichen Fortschritt aller Habits pro Tag zusammenfasst. Die Anzeige des Kalenders basiert auf den gespeicherten entries.

Dateien:

* src/lib/components/CalendarView.svelte
* routes/stats/+page.svelte
* routes/stats/+page.server.js

## Kategorie-System mit eigener Collection:

Ein Bild, das Haushaltsgerät, Gerät, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

Habits sind über ein Dropdown einer Kategorie zugeordnet. Kategorien sind als eigene Collection (categories) in der Datenbank gespeichert. Die Verwaltung erfolgt über die Seite /categories.åå

Dateien:

* lib/db/categories.js
* routes/categories/+page.svelte
* routes/habits/new/+page.svelte
* routes/habits/new/+page.server.js

## Responsive-Layout:

Auf der gesamten Webanwendung wird Bootstrap 5 verwendet und ist dadurch auf Mobilgeräten auch vollständig responsive. Die Navigation ist als einklappbare Mobile-Navbar implementiert.

Dateien:

* src/app.hmtl
* routes/+layout.svelte
* Globale Verwendung von Bootstrap-Klassen (container, row, col, btn usw.)

## Bearbeitungsfunktion für Habits:

Auf der Detailseite eines Habits (/habits/[id]) befindet sich ein Button zum Bearbeiten. Dieser führt zur Seite /habits/[id]/edit, wo die bestehenden Werte vorausgefüllt sind und geändert werden können.

Dateien:

* routes/habits/[id]/+page.svelte
* routes/habits/[id]/edit/+page.svelte
* routes/habits/[id]/edit/+page.server.js