Prototyping Projektdokumentation

Name: Pascal Störrlein

E-Mail: stoerpas@students.zhaw.ch

URL der deployten Anwendung: [https://stoerpaspt.netlify.app/

# Einleitung

Die Anwendung ProGolf Otelfingen soll als eine erste Instanz für den Pro Shop des Golfparks Otelfingen dienen. Dabei sollen Golf Sets und einzelne Schläger zur Auswahl stehen. Zu den Kernfunktionen gehören das Erstellen, Bearbeiten, Löschen und Anzeigen von Golfsets und einzelnen Clubs. Eine MongoDB-Datenbank dient als zentrales Speichermedium, wobei die Medien Dateien in VS Code gespeichert wurden. Für die Golfschläger werden Golf spezifische Daten angezeigt wie Loft und für welche Reifegrad von Spieler die Schläger gedacht sind. Die Golfsets können aus mehreren Golfschlägern zusammengesetzt werden um personalisierte Golfsets zu erstellen. Aktuell konnte noch kein Shop dazu erstellt werden. Ebenso vorhanden ist die aktuelle Wetterlage des Golfplatzes.

Die meiste Darstellung wurde via Bootstrap versucht als auch mit CSS Styling. Wie bereits in den Moviewatchlist verwendete Routing und Navigationsfunktion wurden ebenfalls miteingebaut.

# Datenmodell

# *Ein Bild, das Diagramm, Kreis, Text, Reihe enthält. KI-generierte Inhalte können fehlerhaft sein.*Beschreibung der Anwendung

*[Beschreibung der einzelnen Pages und Funktionen inklusive Screenshots. Die Funktionalität und die Workflows müssen anhand der Screenshots und Textbeschreibungen nachvollziehbar sein. Der Code muss nicht beschrieben werden.]*

## Startseite

**Route:** /  
Hier wird eine einfache Landing Page dargestellt. Navigation erfolgt über die Navbar.

**Dateien:**

* routes/+page.svelte
* routes/+layout.svelte

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

KI-generierte Inhalte können fehlerhaft sein.

## GolfClubs anzeigen

**Route:** /club  
Listet alle Golfclubs in einer Tabelle mit einer Hervorhebung wenn man mit dem Zeiger darüber fährt. Die Komponente GolfClub.svelte stellt einzelne Clubs dar.

**Dateien:**

* routes/club/+page.svelte
* routes/club/+page.server.js
* lib/components/GolfClub.svelte
* lib/db.js

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

KI-generierte Inhalte können fehlerhaft sein.

## GolfClub erstellen

**Route:** /club/create  
Formular zur Eingabe eines neuen Clubs. Für das Image wurde ein Platzhalter erstellt, da ein Bild-Upload leider nicht mehr implementiert werden konnte. Bootstrap-Komponenten wie form-control, btn, mb-3 werden verwendet.

**Dateien:**

* routes/club/create/+page.svelte
* routes/club/create/+page.server.js
* lib/db.js

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

KI-generierte Inhalte können fehlerhaft sein.

## GolfClub in Details ansehen und/oder bearbeiten

**Route:** /club/[club\_id]/edit  
Dynamische Route mit Pfadparameter. Daten werden mit load() geladen, Formular mit vorgefüllten Daten angezeigt.

**Dateien:**

* routes/club/[club\_id]/edit/+page.svelte
* routes/club/[club\_id]/edit/+page.server.js
* lib/db.js

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

KI-generierte Inhalte können fehlerhaft sein.

Für die Typen der Golfschläger existiert eine Drop Down Liste mit den Typen: Driver, Iron und Putter.

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

KI-generierte Inhalte können fehlerhaft sein.

## GolfSets anzeigen und neues Set erstellen

**Route:** /set  
Listet alle Golfsets mit Verknüpfung zur Detailansicht. Verwendung von GolfSet.svelte für Kartenansicht im Grid-Layout (row, col-md-4).

**Dateien:**

* routes/set/+page.svelte
* routes/set/+page.server.js
* lib/components/GolfSet.svelte
* lib/db.js

Ein Bild, das Screenshot, Text, Tasche enthält.

KI-generierte Inhalte können fehlerhaft sein.

Für die Auswahl der Schläger wurde eine Checkbox Liste ausgewählt, wobei es bei der Bearbeitung via Komponenten noch besser aussieht.

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

KI-generierte Inhalte können fehlerhaft sein.

## GolfSet Details

**Route:** /set/[set\_id]  
Zeigt Details eines Sets und die zugehörigen Clubs. Dynamische Daten via load() geladen. Ebenso können kann das Set hierbei bearbeitet oder gelöscht werden.

**Dateien:**

* routes/set/[set\_id]/+page.svelte
* routes/set/[set\_id]/+page.server.js

Ein Bild, das Screenshot, Text, Design enthält.

KI-generierte Inhalte können fehlerhaft sein.

## GolfSet bearbeiten

**Route:** /set/create, /set/[set\_id]/edit  
Form Actions zur Eingabe und Speicherung in MongoDB. Verwendung von bind:value, Formularobjekten und actions.create. Die Schläger können via Komponente und Checkbox für das Set selektiert oder de-selektiert werden.

**Dateien:**

* routes/set/create/+page.svelte
* routes/set/create/+page.server.js
* routes/set/[set\_id]/edit/+page.svelte
* routes/set/[set\_id]/edit/+page.server.js
* lib/db.js

Ein Bild, das Text, Screenshot, Auto enthält.

KI-generierte Inhalte können fehlerhaft sein.

## CSS Styles

Für die verschiedenen Komponenten wurden globale CSS Styles erstellt, für eine einfache Handhabung bei seriöser Ausarbeit. Ein Beispiel für das Design der Schläger:

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

## MongoDB Anbindung

Die Club und Set Informationen konnten via MongoDB auf zwei collections gespeichert werden:

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Die Schläger welche bei einem Set dabei sind, sind jeweils in einem Array als Objekt gespeichert:

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Leider konnten die Bilder und Video nicht auch auf eine flexiblere Art und Weise gespeichert werden. Somit sind sie im static Folder gespeichert:

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

KI-generierte Inhalte können fehlerhaft sein.

# Erweiterungen

*[Liste der Erweiterungen, die über die Grundanforderungen hinausgehen, siehe Punkt «Erweiterungen» in der Aufgabenstellung. Achten Sie auf Vollständigkeit.* ***Es werden nur die hier beschriebenen Erweiterungen bewertet****. Der Code muss grundsätzlich nicht beschrieben werden, aber es sollten Hinweise gegeben werden, wo die Erweiterung implementiert wurde (Dateiname, Funktionsname, etc.). Screenshots sind nur nötig, falls die Erweiterung nicht unter «Beschreibung der Anwendung» beschrieben wurde.]*

*Beispiel:*

## Wetter API

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

KI-generierte Inhalte können fehlerhaft sein.

Es wurde ein Wetterbericht via weatherapi miteingebunden, mit dem Sinne, dass Golfer sehen ob schön Wetter ist oder nicht. Für die API wurde ein neuer Folder unter lib, sowie die weather.js Datei erstellt:

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

KI-generierte Inhalte können fehlerhaft sein.

Eine weitere Erweiterung war das Intro Video auf der Route/+page.svelte, welches anhand einer lokalen videos Instanz in einem Loop abgespielt wird:

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

KI-generierte Inhalte können fehlerhaft sein.

Eine mögliche Erweiterung war es die Golfschläger, als Komponenten für die jeweiligen Sets darzustellen und so die N zu M Beziehung vorzuzeigen:

Ein Bild, das Licht, Design enthält.

KI-generierte Inhalte können fehlerhaft sein.