

Prototyping Projektdokumentation

Name: Jan Voser

E-Mail: voserjan@students.zhaw.ch

URL der deployten Anwendung: <https://gymtracker77.netlify.app>

1. Einleitung

GymTracker ist eine moderne Webanwendung zur Erfassung und Analyse von Krafttrainingseinheiten. Als begeisterter Sportler wollte ich eine Lösung entwickeln, die ich selbst in Zukunft im Trainingsalltag nutzen kann. Die Anwendung ermöglicht es BenutzerInnen, ihre Workouts zu dokumentieren, Fortschritte über die Zeit zu analysieren und gezielt nach Muskelgruppen oder Equipment zu filtern. Ziel war es, ein benutzerfreundliches und visuell ansprechendes Tool zu entwickeln, das eine klare und effiziente Nutzung ermöglicht.

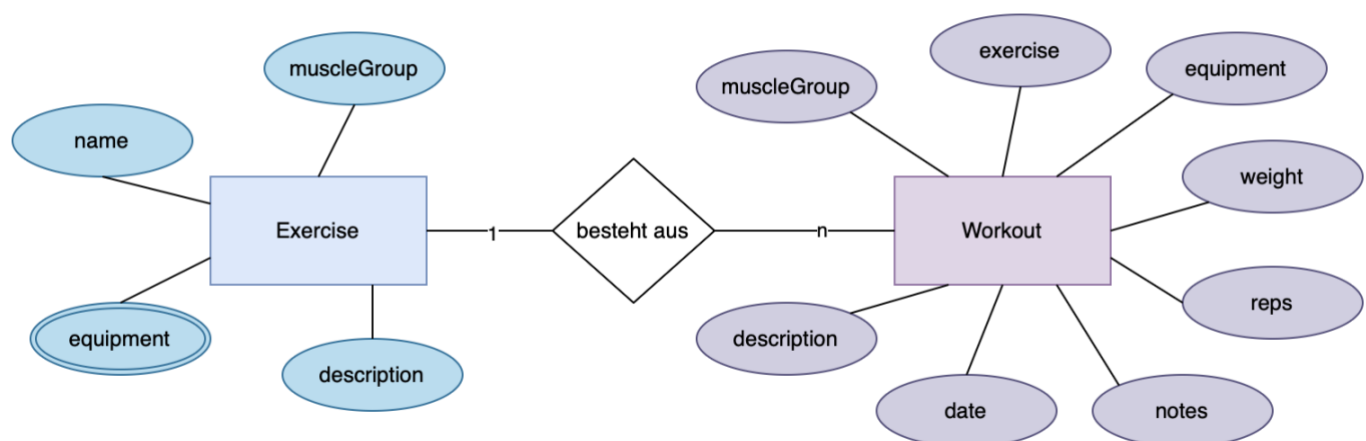
Die Umsetzung erfolgte mit SvelteKit im Runes-Modus, MongoDB als Datenbank sowie D3.js für die Fortschrittsvisualisierung. Das Projekt entstand im Rahmen des Moduls «Prototyping» an der ZHAW und orientiert sich an der Beispielstruktur der «Movie Watchlist».

2. Datenmodell

In der GymTracker-Anwendung existieren zwei zentrale Entitäten: Exercise und Workout. Exercises beschreiben allgemeine Informationen wie Name, Muskelgruppe und verfügbares Equipment. Workouts hingegen erfassen konkrete Trainingseinträge mit Datum, Gewicht, Wiederholungen und weiteren Angaben.

Zwischen den beiden Entitäten besteht eine 1:n-Beziehung: Eine Übung kann mehrfach in verschiedenen Workouts verwendet werden. Ein Workout besteht aus genau ein Exercise. Die Verknüpfung erfolgt über den Namen der Übung, der im Workout gespeichert wird.

2.1. ER-Diagramm



2.2. Beispiel der Entitäten aus MongoDB

Exercise

```
_id: ObjectId('68388929e51abb8082c9966d')
name: "Bankdrücken"
muscleGroup: "Brust"
equipment: Array (3)
  0: "Langhantel"
  1: "KurzhankeIn"
  2: "Maschine"
description: "Klassische Übung für den Aufbau der Brustmuskulatur."
```

Workout

```
_id: ObjectId('683c6715e831ff1e39e63890')
exercise: "Bankdrücken"
equipment: "Langhantel"
weight: 80
reps: "3x10"
notes: ""
date: "2025-06-01"
muscleGroup: "Brust"
description: "Klassische Übung für den Aufbau der Brustmuskulatur."
```

3. Beschreibung der Anwendung

Die Datei +layout.svelte definiert das globale Layout der Applikation, insbesondere die Navigationsleiste. Sie ist auf allen Seiten sichtbar, wird aber im Folgenden nicht bei jeder einzelnen Page separat aufgeführt. Die Beschreibung konzentriert sich jeweils auf die Seiten spezifischen Dateien wie +page.svelte und +page.server.js.

3.1. Startseite

Route:



Beim Aufruf der Startseite wird ein Willkommensscreen angezeigt. Im Zentrum steht eine motivierende Botschaft, ergänzt durch einen prominenten Call-to-Action-Button «Workout starten». Dieser führt direkt zur Seite zum Hinzufügen eines neuen Workouts.

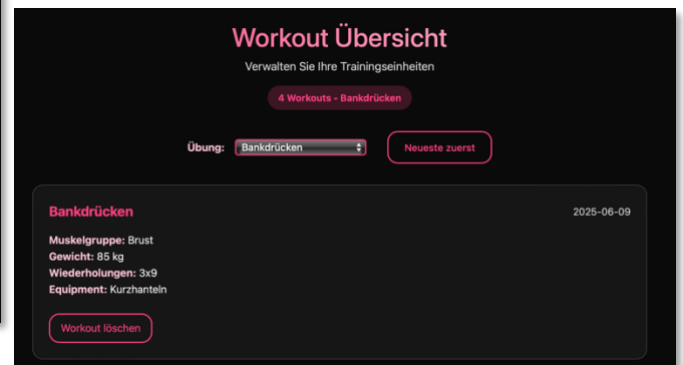
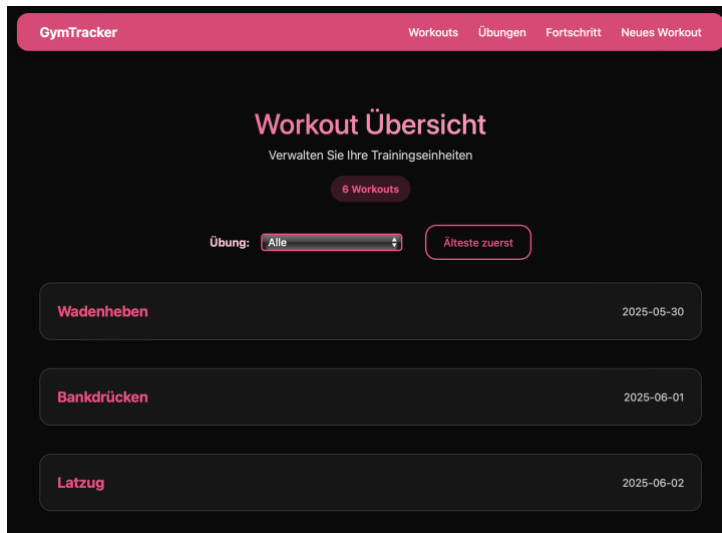
Die Startseite hat keinen weiteren funktionalen Inhalt, dient aber als optisch klarer Einstiegspunkt in die Applikation. Besonders auf Desktop-Bildschirmen wirkt das Hintergrundbild motivierend.

Dateien:

- routes+page.svelte

3.2. Workoutübersicht

Route: /workouts



Auf dieser Seite werden alle bisherigen Workouts tabellarisch dargestellt. Zu Beginn ist eine Zusammenfassung mit Anzahl der erfassten Einträge zu sehen. Die Benutzer können mithilfe eines Dropdown-Menüs gezielt nach Übungen filtern (z. B. nur «Bankdrücken» anzeigen lassen) und zusätzlich die Sortierung nach Datum umschalten.

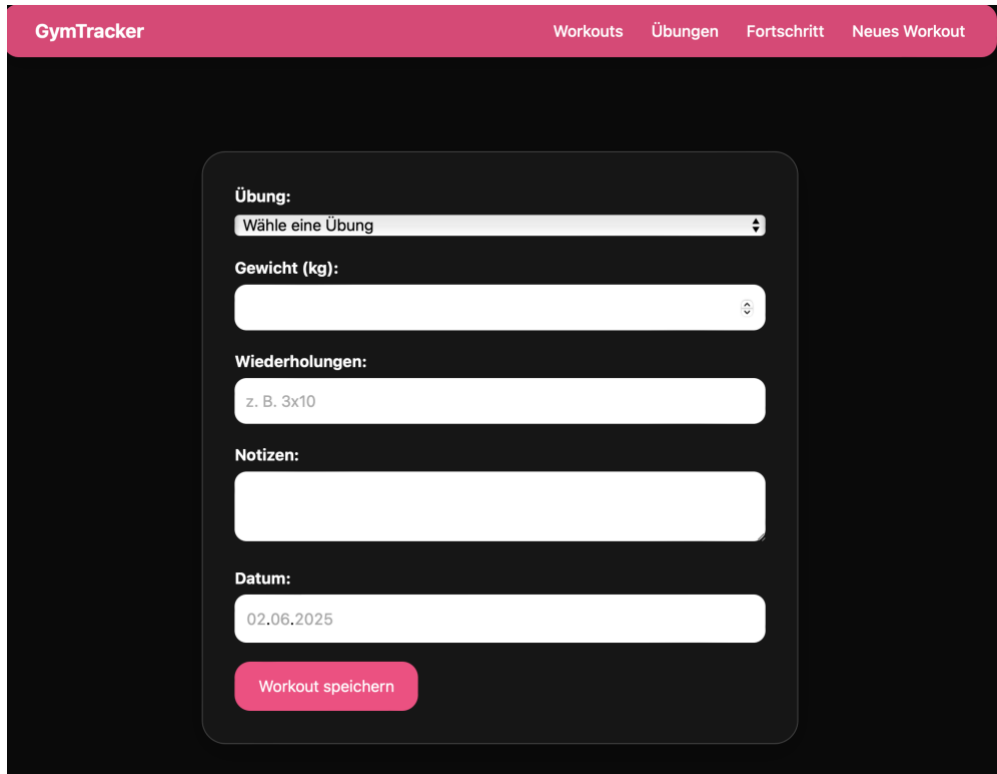
Jedes Workout wird als einzelne Karte angezeigt. Durch Klicken auf die Karte werden zusätzliche Details eingeblendet wie Muskelgruppe, Equipment, Gewicht, Wiederholungen, Notizen und das Trainingsdatum. Über den Button «Workout löschen» kann ein Eintrag entfernt werden. Das Löschen erfolgt direkt über die integrierte Serverroute.

Dateien:

- routes/workouts/+page.svelte
- routes/workouts/+page.server.js
- lib/components/WorkoutCard.svelte
- routes/workouts/delete/[id]/+server.js
- lib/db.js

3.3. Neues Workout erfassen

Route: /workouts/create



The screenshot shows a web application interface for creating a new workout. At the top, there is a pink navigation bar with the text 'GymTracker' on the left and four menu items: 'Workouts', 'Übungen', 'Fortschritt', and 'Neues Workout'. The 'Neues Workout' item is highlighted. Below the navigation bar, the main content area has a dark background. In the center, there is a white rounded rectangle containing a form. The form has the following fields: 'Übung:' with a dropdown menu showing 'Wähle eine Übung'; 'Gewicht (kg):' with a text input field; 'Wiederholungen:' with a text input field showing 'z. B. 3x10'; 'Notizen:' with a larger text input field; and 'Datum:' with a text input field showing '02.06.2025'. At the bottom of the form is a pink button labeled 'Workout speichern'.

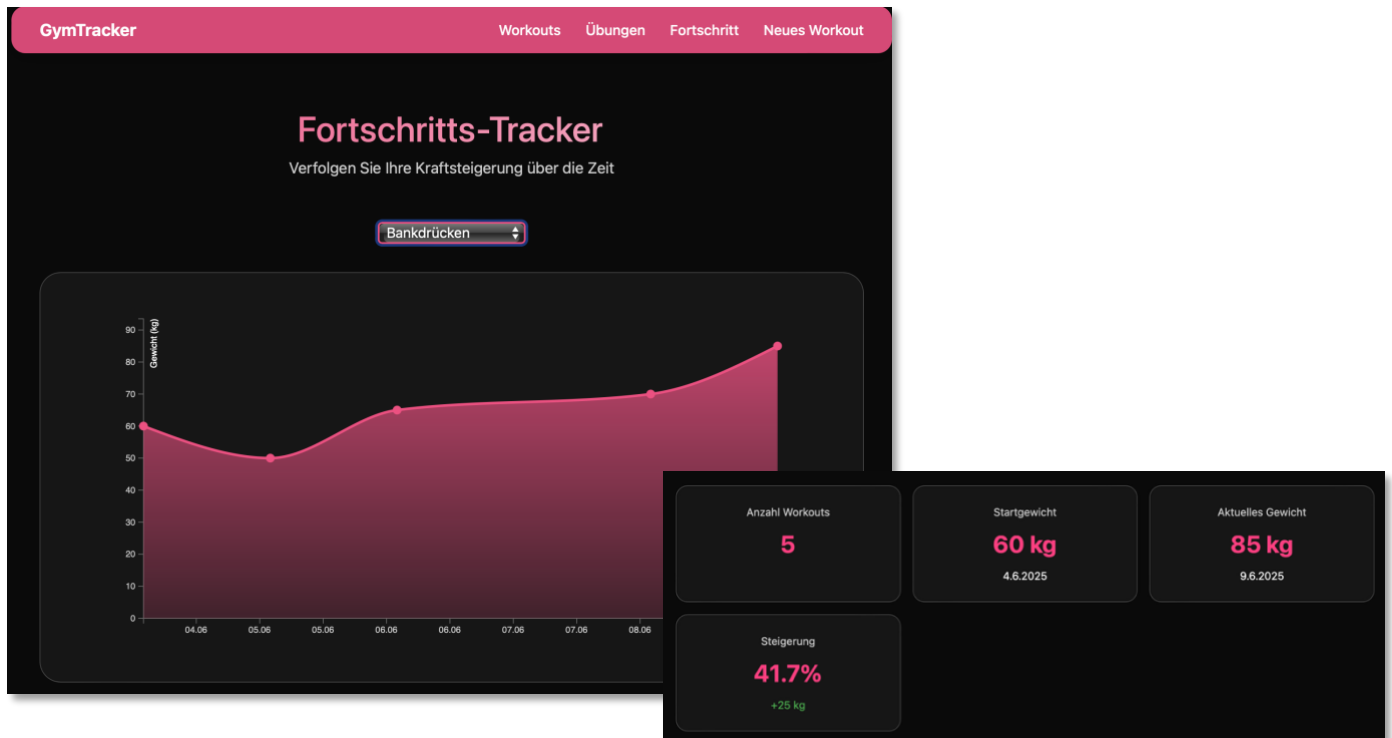
Auf dieser Seite können Benutzer ein neues Workout erfassen. Zuerst wählt man eine Übung aus einer bestehenden Liste aus, welche zuvor in der Datenbank gespeichert wurde. Sobald eine Übung ausgewählt ist, werden Zusatzinformationen wie Muskelgruppe, Beschreibung und verfügbares Equipment automatisch angezeigt. Je nach Übung kann zusätzlich das verwendete Equipment aus einer Dropdown-Liste gewählt werden. Anschliessend werden manuell das Gewicht, die Wiederholungen, Notizen sowie das Trainingsdatum eingegeben. Nach dem Absenden des Formulars wird das Workout in der MongoDB gespeichert und der Benutzer automatisch zur Workoutübersicht weitergeleitet

Dateien:

- routes/workouts/create/+page.svelte
- routes/workouts/create/+page.server.js
- lib/db.js

3.4. Fortschritt anzeigen

Route: /progress



Diese Seite visualisiert die Trainingsentwicklung über die Zeit. Im Dropdown-Menü kann eine spezifische Übung ausgewählt werden. Für die gewählte Übung werden alle zugehörigen Workouts aus der Datenbank geladen und als Zeitreihe dargestellt.

Die Darstellung erfolgt über ein D3.js-Liniendiagramm. Die X-Achse zeigt die Trainingsdaten, die Y-Achse das verwendete Gewicht. Zusätzlich wird die Fläche unter der Kurve farbig hervorgehoben. Beim Überfahren der Datenpunkte erscheinen Tooltips mit Gewicht, Wiederholungen und Datum des Workouts.

Unterhalb des Diagramms werden wichtige Statistiken berechnet und angezeigt:

- Anzahl der absolvierten Workouts
- Startgewicht mit Datum
- Letztes Gewicht mit Datum
- Prozentuale Steigerung

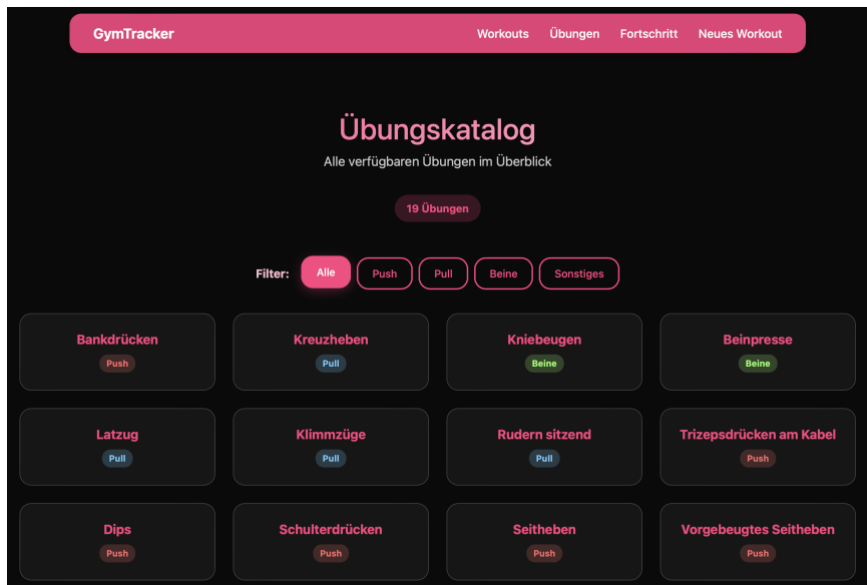
Diese Fortschrittsanzeige hilft den Benutzer, ihre Leistung langfristig zu verfolgen und Entwicklungen sichtbar zu machen.

Dateien:

- routes/progress/+page.svelte
- routes/progress/+page.server.js
- lib/db.js

3.5. Übungskatalog

Route: /exercises



In diesem Bereich der Anwendung werden alle verfügbaren Übungen aufgelistet. Jede Übung ist in einer Exercise Card dargestellt und zeigt den Namen sowie die zugehörige Muskelkategorie (Push, Pull, Beine, Sonstiges). Diese Kategorie wird dynamisch anhand der Muskelgruppe zugewiesen und durch ein farbiges Badge hervorgehoben. Über die Filterleiste oben auf der Seite können Benutzer gezielt nach Kategorien. Beim Klick auf eine Übung wird man auf die Detailseite dieser Übung weitergeleitet.

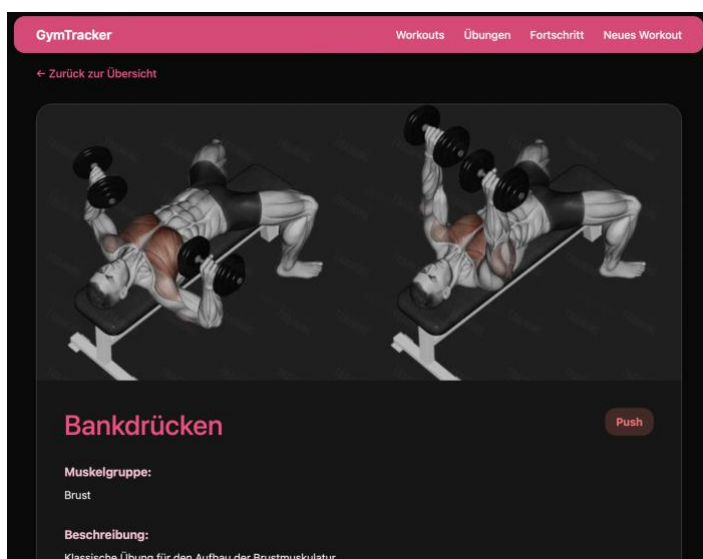
Ziel dieses Bereichs ist es, den Benutzer eine übersichtliche Navigation und Kategorisierung ihrer Übungen zu ermöglichen.

Dateien:

- routes/exercises/+page.svelte
- routes/exercises/+page.server.js
- lib/db.js

3.6. Übungsdetails

Route : /exercises/[id]



Auf der Detailseite einer Übung erhalten Benutzer alle relevanten Informationen zu einer ausgewählten Übung inklusiv passendem Bild.

Dateien:

- routes/exercises/[id]/+page.svelte
- routes/exercises/[id]/+page.server.js
- lib/db.js

4. Erweiterungen

4.1. Fortschrittvisualisierung mit D3.js

Auf der Seite /progress werden Gewichtsentwicklungen einzelner Übungen mit D3.js als interaktives Liniendiagramm visualisiert. Zusätzlich werden Live-Statistiken (Startgewicht, aktuelles Gewicht, Steigerung in %) dynamisch berechnet

Dateien:

- lib/db.js
- routes/progress/+page.svelte
- routes/progress/+page.server.js

4.2. Erweiterte Filterfunktionen

In der Workout-Übersicht /workouts wurden zwei zusätzliche Filter integriert:

- Sortierung (neueste zuerst / älteste zuerst)
- Übungsfilter (z. B. nur «Bankdrücken»)

Dateien:

- routes/workouts/+page.svelte
- routes/workouts/+page.server.js
- lib/db.js

4.3. Responsives Design mit Bootstrap

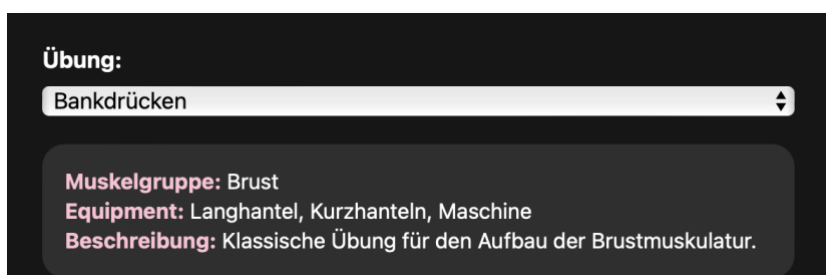
Die Anwendung nutzt Bootstrap 5 zur Gestaltung eines modernen und responsiven Layouts. Grundlegende Komponenten wie Buttons, Container und Layoutbereiche verwenden Bootstrap-typische Klassen für Abstand, Farben und Flexbox-Struktur. Dadurch passt sich das Design grundsätzlich an verschiedene Bildschirmgrößen an.

Dateien:

- Alle +page.svelte Dateien

4.4. Einleitung zu den Erweiterungen 4.4.1 – 4.4.3

Im Rahmen der Erweiterungen wurde der Fokus auf eine dynamisch reagierende und benutzerfreundliche Eingabe bei der Workout-Erstellung gelegt. Nach Auswahl einer Übung werden automatisch weitere relevante Informationen angezeigt und in das Workout übernommen ohne manuelle Eingabe. Diese Verbesserungen erhöhen die Benutzerfreundlichkeit, minimieren Eingabefehler und sorgen für eine effizientere Nutzung der Anwendung.



Übung:

Bankdrücken

Muskelgruppe: Brust

Equipment: Langhantel, Kurzhanteln, Maschine

Beschreibung: Klassische Übung für den Aufbau der Brustmuskulatur.

4.4.1. Dynamische Equipment Auswahl

Bei der Erstellung eines neuen Workouts wird das verwendete Equipment automatisch aus der Übung übernommen und dynamisch angezeigt.

Dateien:

- routes/workouts/create/+page.svelte
- routes/workouts/create/+page.server.js
- lib/db.js

4.4.2. Automatische Muskelgruppenzuordnung

Beim Erstellen eines neuen Workouts wird basierend auf der ausgewählten Übung die zugehörige Muskelgruppe automatisch mit in das Workout übernommen und gespeichert.

Dateien:

- routes/workouts/create/+page.svelte
- routes/workouts/create/+page.server.js
- lib/db.js

4.4.3. Automatische Beschreibungseinbindung

Beim Erstellen eines neuen Workouts wird nach Auswahl einer Übung die zugehörige Beschreibung automatisch angezeigt. Die Beschreibung bietet dem Benutzer eine kurze inhaltliche Einordnung der Übung und dient der besseren Nachvollziehbarkeit bei der Eingabe.

Dateien:

- routes/workouts/create/+page.svelte
- routes/workouts/create/+page.server.js
- lib/db.js

4.5. Bildintegration Detailansicht

In der Detailansicht jeder Übung /exercises/[id] wird ein passendes Bild eingeblendet. Die Bilder werden basierend auf dem Übungsnamen dynamisch aus dem Verzeichnis /images geladen. Zusätzlich wird ein Fallback-Bild angezeigt, falls kein Bild vorhanden ist.

Dateien:

- routes/exercises/[id]/+page.svelte
- static/images/

4.6. Konsistente Navigation

Alle Seiten verwenden ein zentrales Layout +layout.svelte, das eine Navigationsleiste mit Links zu allen Hauptbereichen enthält: Workouts, Übungen, Fortschritt, Neues Workout

Dies sorgt für eine konsistente Benutzerführung und spart redundanten Code.

Dateien:

- routes/+layout.svelte

4.7. Visuelle Rückmeldungen & UX

Buttons heben sich durch Hover-Effekte und Farbwechsel deutlich ab. Erfolgreich gespeicherte Einträge führen direkt zur Übersicht zurück. Löschaktionen werden durch ein Bestätigungsdialog abgesichert.

Dateien:

- `routes/workouts/+page.svelte`
- `lib/components/WorkoutCard.svelte`
- `routes/exercises/+page.svelte`
- `routes/workouts/delete/[id]/+server.js`