

Prototyping Projektdokumentation

Name: Seyid Hussein Hussein

E-Mail: hussesey@students.zhaw.ch

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

1. Einleitung

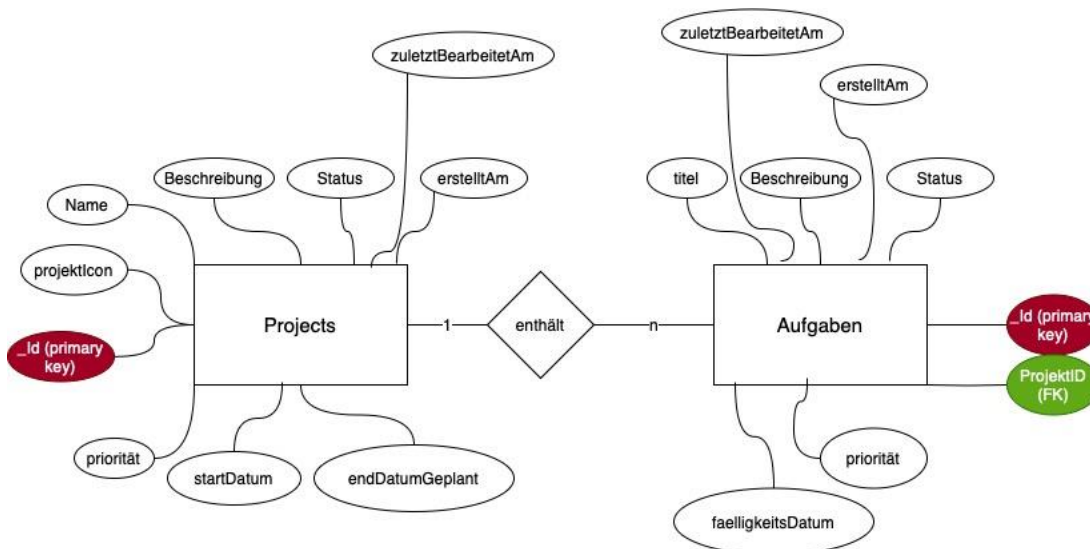
Die Webanwendung "Mein ProjektTracker" ist ein persönliches Tool zur effizienten Organisation und Verwaltung von Projekten und den dazugehörigen Aufgaben. Die Idee hinter der Anwendung ist es, eine klare und intuitive Oberfläche zu bieten, mit der Nutzer ihre Vorhaben strukturieren, deren Fortschritt verfolgen und Deadlines im Blick behalten können. Im hektischen Alltag sei es im Studium oder bei privaten Vorhaben, gehen oft Details unter oder die Übersicht geht verloren. "Mein ProjektTracker" zielt darauf ab, diesem Problem entgegenzuwirken und eine zentrale Anlaufstelle für alle anfallenden Projekte und Aufgaben zu schaffen.

Die Grundfunktionen der Anwendung umfassen:

- Das Erstellen, Anzeigen (in einer Listen- und einer detaillierten Einzelansicht), Bearbeiten und Löschen von Projekten.
- Jedem Projekt können optional visuelle Icons zugeordnet werden, um die Übersichtlichkeit zu erhöhen.
- Zu jedem Projekt können Aufgaben erstellt und wieder gelöscht werden, die spezifisch diesem Projekt zugeordnet sind.
- Eine Dashboard-Ansicht bietet eine schnelle Übersicht über den Status der laufenden Projekte und hebt besonders priorisierte Vorhaben hervor.

Die Anwendung wurde mit Svelte 5 im Runes-Mode, SvelteKit und MongoDB entwickelt, um eine moderne, reaktive und performante Nutzererfahrung zu gewährleisten.

2. Datenmodell



Erläuterungen zum Datenmodell:

Das Datenmodell der Anwendung "Mein ProjektTracker" nutzt MongoDB und besteht aus zwei Hauptentitäten: Projects (Projekte) und Aufgaben. Jedes Dokument erhält eine automatisch generierte _id (ObjectId) als Primärschlüssel (PK).

Entität: Projects (Projekte) Speichert Informationen zu einzelnen Projekten.

Wichtige Attribute:

- `_id` (ObjectId, PK): Eindeutiger Identifikator.
- `name` (String): Pflichtfeld, Name des Projekts.
- `beschreibung` (String, Optional): Detaillierte Beschreibung.
- `status` (String, Pflichtfeld): Aktueller Status (z.B. "Planung", "In Arbeit").
- `prioritaet` (String, Optional): Priorität (z.B. "Hoch", "Mittel").
- `startDatum` (Date, Optional): Projektbeginn.
- `endDatumGeplant` (Date, Optional): Geplantes Projektende.
- `projektIcon` (String, Optional): Dateiname des zugehörigen Icons.
- `erstelltAm` (Date, Automatisch): Erstellungsdatum.
- `zuletztBearbeitetAm` (Date, Automatisch): Datum der letzten Änderung.

Entität: Aufgaben

Speichert Details zu Aufgaben, die einem Projekt zugeordnet sind.

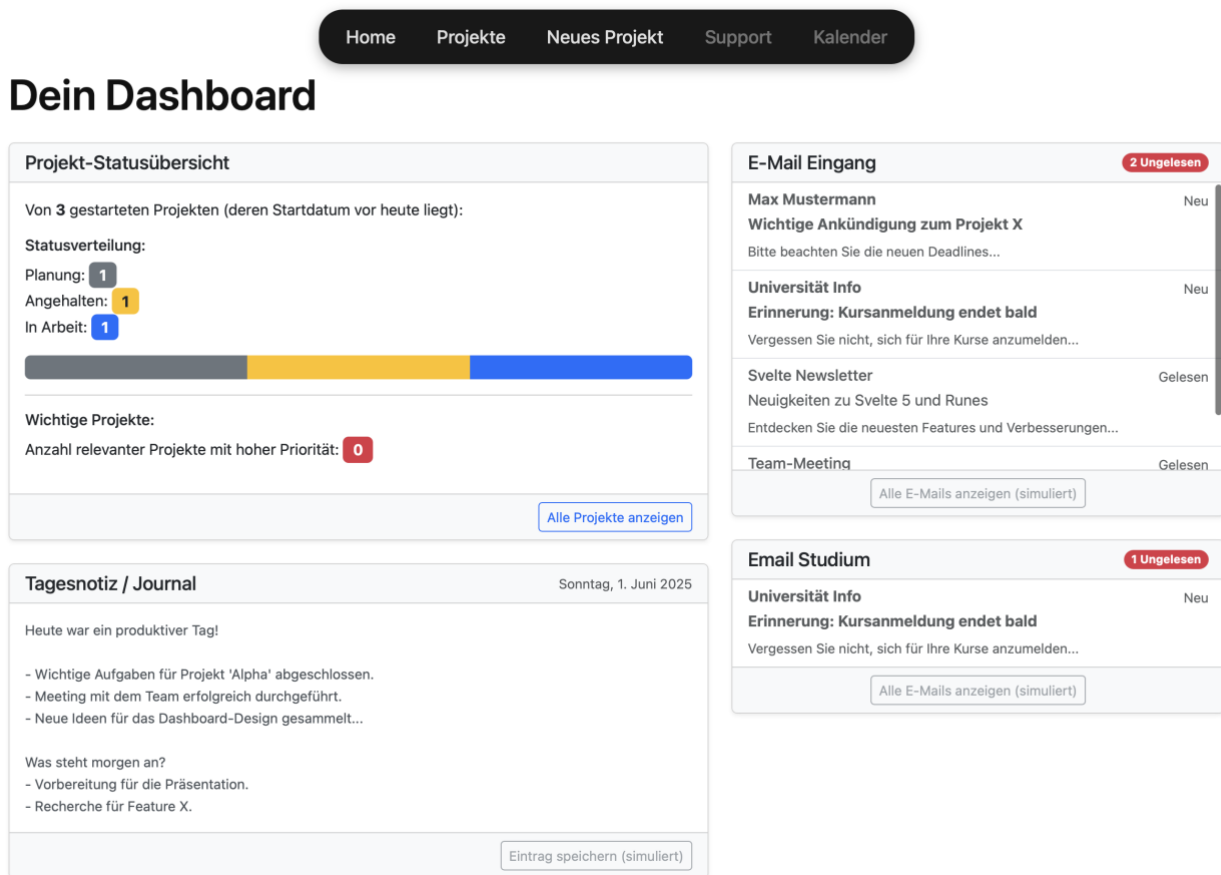
Wichtige Attribute:

- `_id` (ObjectId, PK): Eindeutiger Identifikator.
- `titel` (String, Pflichtfeld): Bezeichnung der Aufgabe.
- `beschreibung` (String, Optional): Details zur Aufgabe.
- `projektId` (ObjectId, FK, Pflichtfeld): Verweis auf die `_id` des zugehörigen Projekts.
- `status` (String, Pflichtfeld): Aktueller Status (z.B. "Offen", "Erledigt").
- `prioritaet` (String, Optional): Priorität der Aufgabe.
- `faelligkeitsDatum` (Date, Optional): Bis wann die Aufgabe erledigt sein soll.
- `erstelltAm` (Date, Automatisch): Erstellungsdatum.
- `zuletztBearbeitetAm` (Date, Automatisch): Datum der letzten Änderung.

3. Beschreibung der Anwendung

3.1. Dashboard

Route: /



Das Dashboard dient als zentrale Startseite der Anwendung "Mein ProjektTracker" und bietet dem Nutzer eine schnelle Übersicht über seine Projekte und weitere illustrative Widgets. Nach der Navigation wird der Titel "Dein Dashboard" unterhalb der Hauptnavigationsleiste angezeigt. Die Seite ist in einem zweispaltigen Layout aufgebaut:

Linke Spalte:

Projekt-Statusübersicht: Dieses Widget zeigt dynamisch geladene Statistiken zu den Projekten an, deren Startdatum vor dem aktuellen Tag liegt. Es visualisiert die Verteilung der Projekte auf verschiedene Status (z.B. Planung, In Arbeit) mittels Zähler und einem gestapelten Balken. Zusätzlich wird die Anzahl der gestarteten Projekte mit hoher Priorität angezeigt. Ein Link führt zur detaillierten Projektübersichtsseite (/projekte).

Tagesnotiz / Journal: Dieses Widget dient der Veranschaulichung und zeigt das aktuelle Datum sowie einen statischen Platzhaltertext für Journaleinträge an. Der Button "Eintrag speichern" ist ein nicht funktionaler Platzhalter.

Rechte Spalte:

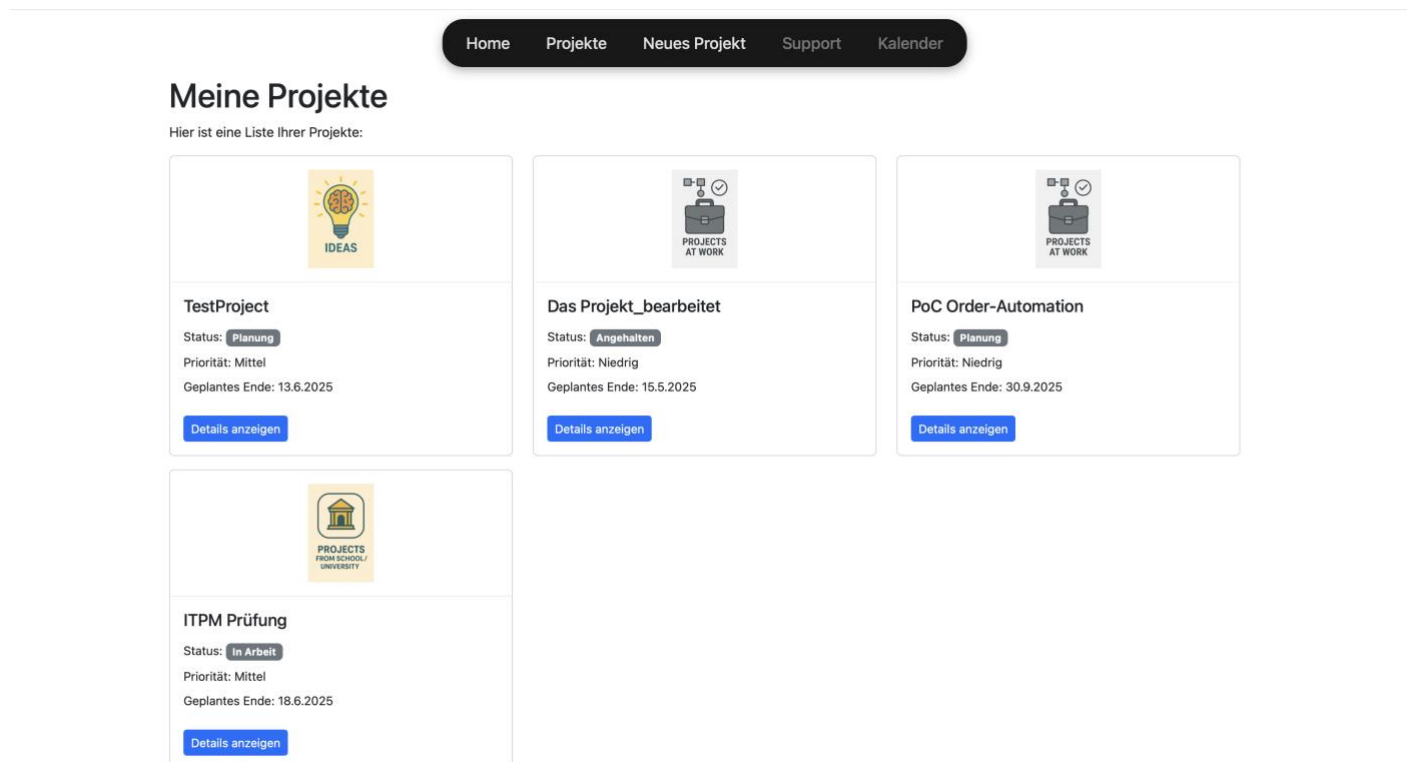
E-Mail-Eingang & E-Mail-Studium: Diese beiden Widgets dienen ebenfalls der Veranschaulichung und zeigen eine Liste von simulierten E-Mails mit Absender. Die Hauptnavigation am oberen Rand ermöglicht den Wechsel zu anderen Bereichen wie der Projektübersicht oder dem Formular zum Erstellen neuer Projekte.

Dateien:

- src/routes/+page.svelte
- src/routes/+page.server.js
- src/lib/components/ProjektDashboardWidget.svelte
- src/lib/components/EmailWidget.svelte
- src/lib/components/DailyJournalWidget.svelte

3.2. Projektübersicht

Route: /projekte



Die Seite "/projekte" dient als zentrale Übersicht aller erfassten Projekte. Nach der Navigation wird der Titel "Meine Projekte" angezeigt. Jede Projektkarte zeigt wesentliche Informationen wie den Projektnamen, das zugehörige Icon, den Status, die Priorität und das geplante Enddatum. Zudem bietet jede Karte einen direkten Link "Details anzeigen" zur jeweiligen Projektdetailseite. Am unteren Rand der Seite befindet sich ein Button "Neues Projekt erstellen", der den Benutzer zum Formular für die Erstellung eines neuen Projekts (/projekte/erstellen) führt.

Dateien:

- src/routes/projekte/+page.svelte
- src/routes/projekte/+page.server.js
- src/lib/components/ProjektKarte.svelte

3.3. Projekt erstellen

Route: /projekte/erstellen

[Home](#) [Projekte](#) [Neues Projekt](#) [Support](#) [Kalender](#)

Neues Projekt erstellen

Projektname *

Beschreibung


Status * Planung ▼


Priorität Mittel ▼


Startdatum 06/01/2025


Geplantes Enddatum 06/01/2025

Projekt-Icon auswählen (optional)

☐  Arbeit

☐  Idee

☐  Privat

☐  Studium

[Abbrechen](#) [Projekt erstellen](#)

© 2025 Mein ProjektTracker

Diese Seite ermöglicht dem Benutzer das Anlegen eines neuen Projekts. Nach dem Ausfüllen der Felder und Klick auf "Projekt erstellen" werden die Daten serverseitig validiert. Bei erfolgreicher Validierung und Speicherung in der Datenbank erscheint ein Pop-up-Fenster mit der Meldung "Projekt erfolgreich erstellt!". Nach Bestätigung des Pop-ups wird der Benutzer automatisch zur Projektübersichtsseite weitergeleitet. Ein "Abbrechen"-Button ermöglicht es dem Benutzer, den Vorgang zu verlassen und zur Projektübersichtsseite zurückzukehren.

Dateien:

- `src/routes/projekte/erstellen/+page.svelte`
- `src/routes/projekte/erstellen/+page.server.js`

3.4. Projekt Detailseite

Route: /projekte/[projektId]

Home


Projekte

Neues Projekt

Support

Kalender

PoC Order-Automation



Beschreibung

Proof of concept of neu Application for getting the orders from emails

Status: Planung **Priorität:** Niedrig

Startdatum: 11.06.2025 **Geplantes Enddatum:** 30.09.2025

Erstellt: 01.06.2025, 01:26
Zuletzt bearbeitet: 01.06.2025, 01:26

Aktionen

Projekt bearbeiten

Zurück zur Übersicht

Projekt endgültig löschen

Zugehörige Aufgaben

+ Neue Aufgabe

Besprechung Intern

Status: In Bearbeitung Priorität: Hoch Fällig bis: 03.06.2025

Löschen

Diese Seite präsentiert alle wichtigen Informationen zu einem ausgewählten Projekt. Der Projektname wird als prominente Überschrift angezeigt. Details wie das Projekt-Icon, die Beschreibung, der aktuelle Status sowie die Priorität sind übersichtlich dargestellt. Ein separater Abschnitt widmet sich den zugehörigen Aufgaben dieses Projekts. Diese werden mit Titel, Status und Fälligkeitsdatum aufgelistet. Jede Aufgabe kann nach einer Sicherheitsabfrage direkt aus der Liste gelöscht werden. Ein Button "+ Neue Aufgabe" führt zum Formular, um dem Projekt eine weitere Aufgabe hinzuzufügen. Über eine seitliche Aktionsleiste kann der Nutzer das angezeigte Projekt bearbeiten, zur allgemeinen Projektübersicht zurückkehren oder das gesamte Projekt nach einer Bestätigungsabfrage löschen. Die für die Anzeige benötigten Daten werden beim Aufruf der Seite geladen.

Dateien:

- src/routes/projekte/[projektId]/+page.svelte
- src/routes/projekte/[projektId]/+page.server.js
- src/lib/server/db.js

3.5. Projekt bearbeiten

Route: /projekte/[projektId]/bearbeiten

[Home](#) [Projekte](#) [Neues Projekt](#) [Support](#) [Kalender](#)

Projekt bearbeiten

PoC Order-Automation

Projektname *

Beschreibung

Status * Priorität

Planung ▼


Niedrig ▼


Startdatum Geplantes Enddatum


06/11/2025


09/30/2025

Projekt-Icon auswählen (optional)

☒  Arbeit

☐  Idee

☐  Privat

☐  Studium

© 2025 Mein ProjektTracker

Diese Seite ermöglicht die Modifikation eines bereits bestehenden Projekts. Beim Aufruf wird das Formular mit den aktuellen Daten des ausgewählten Projekts vorausgefüllt, welche serverseitig geladen werden. Der Benutzer kann die gewünschten Änderungen in den Feldern vornehmen. Nach Klick auf "Änderungen speichern" werden die Daten serverseitig validiert. Bei Erfolg erscheint ein Pop-up-Fenster mit der Meldung "Projekt erfolgreich aktualisiert!", und der Benutzer wird zur Detailansicht des bearbeiteten Projekts zurückgeleitet, wo die aktualisierten Daten sichtbar sind. Treten Validierungsfehler auf, werden diese direkt im Formular angezeigt, und die bisherigen Eingaben bleiben erhalten. Mit dem "Abbrechen"-Button kann der Bearbeitungsvorgang verworfen und zur Projektdetailseite zurückgekehrt werden.

Dateien:

- `src/routes/projekte/[projektId]/bearbeiten/+page.svelte`
- `src/routes/projekte/[projektId]/bearbeiten/+page.server.js`

3.6. Neue Aufgabe erstellen

Route: /projekte/[projektId]/aufgaben/erstellen

[Home](#) [Projekte](#) [Neues Projekt](#) [Support](#) [Kalender](#)

Neue Aufgabe erstellen

Für Projekt: **Das Projekt_bearbeitet**

Titel *

Beschreibung

Status * **Priorität**

Offen Mittel

Fälligkeitsdatum

06/01/2025

[Abbrechen](#) [Aufgabe erstellen](#)

© 2025 Mein ProjektTracker

Über diese Seite können Benutzer neue Aufgaben zu einem spezifischen Projekt hinzufügen. Das Formular selbst bietet Eingabefelder für den Titel der Aufgabe, eine Beschreibung, den Status, die Priorität und ein optionales Fälligkeitsdatum. Nach dem Absenden des Formulars mit "Aufgabe erstellen" werden die Eingaben serverseitig validiert. Bei erfolgreicher Validierung und Speicherung der Aufgabe in der Datenbank erscheint ein Pop-up-Fenster mit der Meldung "Aufgabe erfolgreich erstellt!". Anschliessend wird der Benutzer zur Detailansicht des Projekts zurückgeleitet, wo die neu erstellte Aufgabe in der Aufgabenliste des Projekts erscheint. Ein "Abbrechen"-Button ermöglicht es, den Vorgang zu verlassen und zur Projektdetailseite zurückzukehren. Die notwendigen Projektinformationen (ID und Name) für die Seite werden serverseitig geladen.

Dateien:

- src/routes/projekte/[projektId]/aufgaben/erstellen/+page.svelte
- src/routes/projekte/[projektId]/aufgaben/erstellen/+page.server.js

4. Erweiterungen

4.1. Dashboard-Seite mit dynamischen Projektstatistiken

Die Startseite der Anwendung (/) dient als Dashboard und bietet eine dynamische Übersicht der Projekte. Es werden nur Projekte berücksichtigt, deren Startdatum in der Vergangenheit liegt. Das Dashboard visualisiert die Verteilung der Projekte auf verschiedene Status (mittels Zählern und einem gestapelten Fortschrittsbalken) und zeigt die Anzahl der Projekte mit "Hoher Priorität" an.

Implementierungshinweise:

Die Filterung der Projekte nach Startdatum sowie die Aggregation der Statistikdaten (Statuszählungen, Anzahl hochpriorisierter Projekte) wurde in der load-Funktion der Dashboard-Seite realisiert. Die Darstellung dieser Statistiken erfolgt durch eine dedizierte Komponente.

Dateien:

- `src/routes/+page.server.js` (speziell die load-Funktion)
- `src/lib/components/ProjektDashboardWidget.svelte`
- `src/routes/+page.svelte` (Einbindung des Widgets)

4.2. Illustrative Dashboard-Widgets (E-Mail & Journal)

Zur visuellen Vervollständigung des Dashboard-Konzepts wurden Widgets für "E-Mail-Eingang", "Email-Studium" und "Tagesnotiz / Journal" implementiert. Diese Widgets zeigen statische Beispieldaten und dienen der Veranschaulichung eines möglichen erweiterten Funktionsumfangs sowie einer ansprechenden Dashboard-Gestaltung. Sie verfügen über keine Backend-Anbindung oder dynamische Funktionalität zum Abrufen oder Speichern von E-Mails bzw. Journaleinträgen.

Umsetzungshinweise: Die Widgets wurden als separate Svelte-Komponenten mit vordefiniertem Inhalt und Styling erstellt. Die Darstellung von Beispiel-E-Mails und eines Beispiel-Journaleintrags erfolgt direkt im Code der jeweiligen Komponente.

Dateien:

- `src/lib/components/EmailWidget.svelte`
- `src/lib/components/DailyJournalWidget.svelte`
- `src/routes/+page.svelte` (Einbindung der Widgets)

4.3. Projekt-Icons

Jedem Projekt kann optional ein visuelles Icon aus einer vordefinierten Liste zugeordnet werden. Dies dient der besseren visuellen Unterscheidung und Personalisierung der Projekte in der Anwendung. Die ausgewählten Icons werden sowohl in der Projektübersicht (auf den Projektkarten) als auch in der Projektdetailansicht angezeigt. Die Auswahl des Icons erfolgt beim Erstellen oder Bearbeiten eines Projekts.

Umsetzungshinweise: Das Datenmodell der Projekt-Entität wurde um ein optionales Feld `projektIcon` (String) erweitert, das den Dateinamen des ausgewählten Icons speichert. Die Bilddateien für die Icons sind im Ordner `static/images/projekt-icons/` abgelegt. Die Auswahlmöglichkeit wurde in die Formulare für das Erstellen und Bearbeiten von Projekten integriert.

Dateien:

- `src/lib/server/db.js`
- `src/routes/projekte/erstellen/+page.svelte` (Icon-Auswahl im Formular)
- `src/routes/projekte/[projektId]/bearbeiten/+page.svelte` (Icon-Auswahl im Formular)
- `src/lib/components/ProjektKarte.svelte` (Anzeige des Icons)
- `src/routes/projekte/[projektId]/+page.svelte` (Anzeige des Icons)