

# Prototyping Projektdokumentation

Name: Vasin Kevin Bui

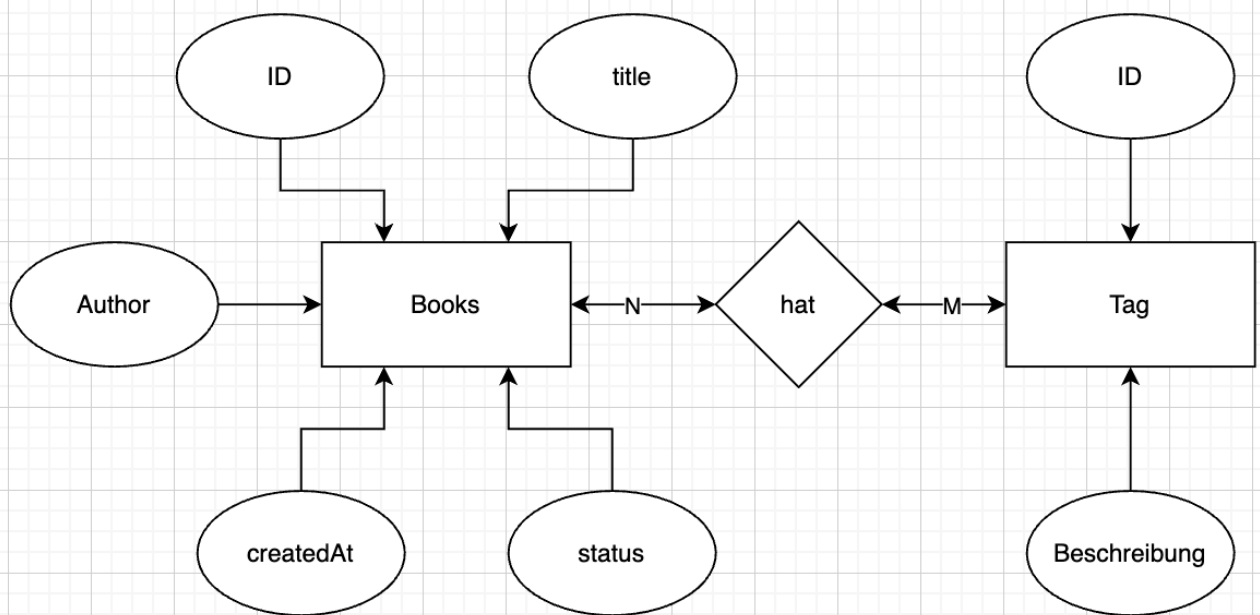
E-Mail: buivas01@students.zhaw.ch

URL der deployten Anwendung: [https://<appname>.netlify.app/]

## 1. Einleitung

Die Leseverwaltungs-App ist eine Webanwendung, mit der Nutzer ihre Bücher digital verwalten können. Sie ermöglicht das Erfassen von gelesenen, offenen und aktuell gelesenen Büchern. Zusätzlich lassen sich Bücher mit Genres (Tags) versehen, um sie später besser zu filtern oder zu suchen.

## 2. Datenmodell



- Books speichert alle Buchdaten inkl. Titel, Autor, Status und Referenzen auf Tags.
- Tags beinhaltet verschiedene Genres wie «Roman», «Thriller» oder «Klassiker». Ein Buch kann mehrere Tags haben, was durch ein Array von ObjectId's (tagIds) abgebildet wird.

## 3. Beschreibung der Anwendung

### 3.1. Startseite

Route: /(Startseite)



## Willkommen bei der Leseverwaltung

Organisiere deine Bücher, verfolge deinen Lesestatus und entdecke neue Klassiker der Literaturwelt.







### Zuletzt hinzugefügt

- verlorenes gold von xatar
- Gari von Gari
- Verlorenes Gold von Xatar



### Deine Statistik

-  Gesamt: **110**
-  Gelesen: **5**
-  Am Lesen: **0**
-  Offen: **105**

Beim Aufrufen der Webanwendung landet man auf der Startseite der Leseverwaltungs-App. Diese dient als zentrale Übersicht und Einstiegspunkt für die wichtigsten Funktionen der Anwendung.

Sie enthält eine kurze Einführung zur Anwendung sowie interaktive Bereiche zur Navigation. Die Benutzer:innen können direkt über Buttons zu den jeweiligen Unterseiten wechseln

- **Neues Buch hinzufügen** – führt zur Seite `/books/new`
- **Alle Bücher anzeigen** – leitet weiter zu `/books`
- **Tags & Genres durchsuchen** – führt zur Übersicht unter `/tags`
- **Suchen nach Titel oder Autor** – ermöglicht direktes Filtern ohne neue Seite
- **Zuletzt hinzugefügte Bücher** – zeigt eine kleine Vorschau der letzten Einträge
- **Statistiken** – gibt einen schnellen Überblick über alle Bücher, unterteilt nach Status
- **Zufälliger Buchtipp** – zeigt bei jedem Laden der Seite ein zufälliges Buch aus der Datenbank an (sofern vorhanden) und motiviert zur Entdeckung neuer Titel

Navigationsleiste (bereitgestellt durch `+layout.svelte`)

Am oberen Rand der Seite befindet sich eine dauerhaft sichtbare Navigationsleiste, über die folgende Seite erreichbar sind:

- Home (`/`)
- Bücher (`/books`)
- Neues Buch (`/books/new`)
- Tags (`/tags`)
- Über (`/about`)

Diese Navigationsleiste bleibt auf allen Unterseiten konsistent sichtbar und erleichtert den Wechsel zwischen den Hauptbereichen der App.


Dateien:

- routes/+page.svelte
- routes/+layout.svelte


### 3.2. Bücherübersicht

Route: /books

 **Leseverwaltung** [Home](#) [Bücher](#) [Neues Buch](#) [Tags](#) [Über](#)



## Deine Bücherliste

 **Filtern nach Status**

☒ Alle

☐ Noch offen

☐ Am lesen

☒ Gelesen

- [1984](#) von George Orwell 

ffnen ▾

Löschen
- [To Kill a Mockingbird](#) von Harper Lee 

Noch offen ▾

Löschen
- [Pride and Prejudice](#) von Jane Austen 

Noch offen ▾

Löschen
- [The Great Gatsby](#) von F. Scott Fitzgerald 

Noch offen ▾

Löschen
- [Moby-Dick](#) von Herman Melville 

Noch offen ▾

Löschen
- [War and Peace](#) von Leo Tolstoy 

Noch offen ▾

Löschen
- [Crime and Punishment](#) von Fyodor Dostoevsky 

Noch offen ▾

Löschen
- [The Catcher in the Rye](#) von J.D. Salinger 

Noch offen ▾

Löschen
- [The Hobbit](#) von J.R.R. Tolkien 

Noch offen ▾

Löschen
- [Brave New World](#) von Aldous Huxley 

Noch offen ▾

Löschen
- [The Brothers Karamazov](#) von Fyodor Dostoevsky 

Noch offen ▾

Löschen
- [Jane Eyre](#) von Charlotte Brontë 

Noch offen ▾

Löschen
- [Wuthering Heights](#) von Emily Brontë 

Noch offen ▾

Löschen
- [Les Misérables](#) von Victor Hugo 

Noch offen ▾

Löschen

Hier werden alle Bücher der Datenbank gelistet.

Funktionen:


- **Filter nach Status** - Nutzer:innen können per Dropdown nach «offen», «am Lesen» oder «gelesen» filtern.
- **Status ändern** - Der Status eines Buches kann direkt angepasst werden.
- **Löschen** - Bücher lassen sich per Button dauerhaft entfernen.

Dateien:

- Routes/books/+page.svelte
- Routes/books/+page.server.ts
- Routes/books/[id]/+server.ts

### 3.3. Buch hinzufügen


Route: /books/new

 **Leseverwaltung** [Home](#) [Bücher](#) [Neues Buch](#) [Tags](#) [Über](#)

## + Neues Buch hinzufügen

Titel:

Autor:

Status:  Noch offen ▼

**Tags**

- ☐ Klassiker
- ☐ Roman
- ☐ Fantasy
- ☐ Science Fiction
- ☐ Krimi
- ☐ Thriller
- ☐ Liebesroman
- ☐ Historisch
- ☐ Sachbuch
- ☐ Biografie
- ☐ Kinderbuch
- ☐ Jugendbuch
- ☐ Horror
- ☐ Abenteuer
- ☐ Drama
- ☐ Comedy
- ☐ Philosophie
- ☐ Psychologie
- ☐ Ratgeber
- ☐ Religion
- ☐ Politik

Speichern

Auf dieser Seite kann ein neues Buch in die Datenbank eingefügt werden.

Felder:

- Titel, Autor, Status
- Tags (Genres) per Checkbox auswählbar

Nach dem Absenden wird das Buch gespeichert und die Seite /books geöffnet.

Dateien:

- Routes/books/new/+page.svelte
- Routes/books/new/+page.server.ts
- Routes/books/new/+server.ts

### 3.4. Buchdetails

Route: /books/[id]



Einzelne Bücher lassen sich über ihre ID anzeigen.

Dargestellt werden:

- Titel, Autor, Status, Erstellungsdatum
- Zugehörige Tags (anklickbar)

Dateien:

- Routes/books/[id]/+page.svelte

- `Routes/books/[id]/+page.server.ts`

### 3.5. Tags (Genre) Übersicht

Route: `/tags`



Hier werden alle gespeicherten Tags (Genres) aufgelistet.

Per Klick auf einen Tag gelangt man zur gefilterten Ansicht der Bücher mit diesem Tag.

Dateien:

- `Routes/tags/+page.svelte`
- `Routes/tags/+page.server.ts`

### 3.6. Bücher mit spezifischem Tag

Route: `/tags/[id]`



## Bücher mit Tag: *Politik*

- [verlorenes gold](#) von xatar

Zeigt eine Liste aller Bücher, die mit dem angeklickten Tag verknüpft sind.

Die Seite wird automatisch durch den Tagnamen generiert.

Dateien:

- Routes/tags/[tag]/+page.svelte
- Routes/tags/[tag]/+page.server.ts

### 3.7. Über-Seite

Route: /about



## Über uns

Willkommen bei **Leseverwaltung** – deiner persönlichen Plattform, um Bücher zu erfassen, zu verwalten und wiederzufinden.

Ob Roman, Sachbuch oder Klassiker: Mit unserer App behältst du stets den Überblick über deine gelesenen Werke, Wunschlisten und Lieblingsautoren.

Unser Ziel ist es, das Lesen nicht nur zu dokumentieren, sondern auch zu inspirieren. Mit einer einfachen Oberfläche, einer Suchfunktion und der Möglichkeit, neue Bücher zu entdecken, möchten wir deine Leseerfahrung digital begleiten.

Dieses Projekt wurde im Rahmen einer Studienarbeit entwickelt und verbindet moderne Webtechnologie (SvelteKit & MongoDB) mit der Liebe zur Literatur.

Viel Spaß beim Verwalten und Entdecken deiner Bücher!

Statische Informationsseite über das Projekt, seine Motivation und Technologien.

Kein interaktiver Inhalt

Dateien:

- Routes/about/+page.svelte

## 4. Erweiterungen

### 4.1. Tag-basiert Kategorisierung und Filterung

In der Anwendung können Bücher mit einem oder mehreren Tags versehen werden (z.B. «Klassiker», «Biografie»). Diese Tags können über die Seite /tags aufgelistet und durch Klick gefiltert werden. So lassen sich alle Bücher mit einem bestimmten Genre anzeigen.

Technik:

Die Tags werden in einer eigenen MongoDB-Collection gespeichert. Beim Erstellen eines Buches kann man mehrere Tags auswählen, deren ObjectIds gespeichert werden. Bei der Anzeige werden die Namen per Join geladen.

Dateien:

- Src/lib/server/db.ts -> getTags(), getBooks(), getBookById()
- Routes/books/new/+page.svelte
- Routes/tags/+page.svelte
- Routes/tags/[tag]/+page.server.ts, +page.svelte

### 4.2. Zufälliger Buchtipp auf der Startseite

Beim Öffnen der Startseite wird zufällig ein Buch ausgewählt und als Buchtipp angezeigt. Das regt Nutzer:innen zur Entdeckung neuer Inhalte an.

Technik:

Ein zufälliger Eintrag aus der books-Liste wird per Math.random() ausgewählt und im Template angezeigt.

Dateien:

- Routes/+page.server.ts -> Auswahl per const random = ...
- Routes/+page.svelte -> Anzeige im Block «Buchtipp»

### 4.3. Live-Suche nach Titel oder Autor

Auf der Startseite kann live nach einem Buchtitel oder Autor gesucht werden. Ergebnisse erscheinen sofort beim Tippen, ohne Neuladen der Seite.

Technik:

Die Funktion nutzt onMount() und ein derived Store, der das Eingabefeld überwacht und clientseitig filtert.

Dateien

- Routes /+page.svelte

### 4.4. Status-Änderung und Filter auf der Buchübersicht

In der Übersicht /books lassen sich Bücher per Dropdown-Feld direkt in ihrem Status ändern (z.B. von «offen» zu «gelesen»). Ausserdem kann man nach Status filtern.

Technik:

- PATCH-Request an /books/[id] beim Ändern des Status
- Filterung über einen lokalen selectedFilter Store

Dateien:

- Routes/books/+page.svelte



- `Routes/books/[id]/+server.ts`