

# Prototyping Projektdokumentation

Name: Thajeena Thiyagarajah

E-Mail: thiyatha@students.zhaw.ch

URL der deployten Anwendung: <https://yourbook-pt.netlify.app/>

## 1. Einleitung

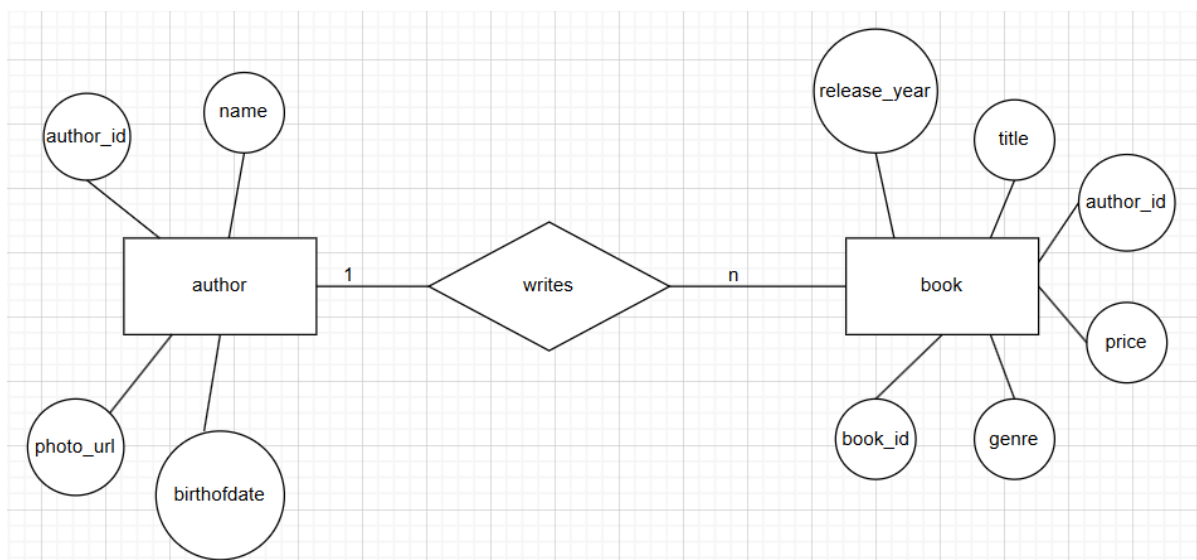
Die Anwendung „yourbook“ ist eine moderne Web-App zur Verwaltung und Sammlung interessanter Bücher. Ziel der Anwendung ist es, eine persönliche „Readlist“ zu erstellen wie eine Merkliste auf der Nutzer Bücher speichern können, die sie später lesen oder wiederfinden möchten.

Auf der Seite „Books“ wird eine Auswahl an Büchern aus einer Datenbank angezeigt mit Titel, Genre, Autor, Preis und Coverbild. Zu jedem Buch stehen zwei Aktionen zur Verfügung. „Details“ öffnet eine Seite mit ausführlichen Informationen zum Buch, während „Readlist“ das Buch zur persönlichen Liste hinzufügt. Zusätzlich können Benutzer über ein Eingabeformular eigene Bücher hinzufügen, welche direkt in der „Readlist“ erscheinen.

Die „Readlist“ selbst ist jederzeit über den Navigationspunkt erreichbar, wird lokal im Browser gespeichert über localStorage und kann durch Entfernen von Einträgen flexibel bearbeitet werden. Das Layout der Seite ist vollständig responsiv gestaltet und funktioniert zuverlässig auf Desktop- und Mobilgeräten. Die Anwendung wurde mit SvelteKit entwickelt, nutzt Form Actions zur Eingabe neuer Daten und bindet Bootstrap für ein Layout und modernes Design ein. Die Navigation ist klar strukturiert und jede Seite ist über das integrierte Routing-System schnell erreichbar.

## 2. Datenmodell

Das Datenmodell meiner Webanwendung basiert auf zwei zentralen Entitäten, Author und Book. Diese bilden die Hauptbestandteile der Anwendung, welche die Verwaltung von Autoren und deren Büchern ermöglicht. Die Entität Author umfasst die Attribute author\_id (eindeutiger Identifikator), name (Name des Autors), birth\_of\_date (Geburtsdatum) und photo\_url (Link zu einem Autorenbild). Die Entität Book beinhaltet die Attribute book\_id (eindeutiger Identifikator), title (Buchtitel), genre (Genre des Buches), price (Preis) und release\_date (Erscheinungsdatum). Zwischen den beiden Entitäten besteht eine 1:n-Beziehung „writes“, da ein Autor mehrere Bücher schreiben kann, jedes Buch jedoch nur einem Autor zugeordnet ist.



### 3. Beschreibung der Anwendung

#### 3.1. Homepage

**Route: /**

Die Startseite der Anwendung dient als Willkommenseite und Einstiegspunkt für Benutzer. Sie besteht aus einem zentralen Layout mit einem Titel, einer kurzen Beschreibung und einem direkten Link zur Buchübersicht unter /books. Ziel ist es, den Nutzer auf die Funktionen der Anwendung aufmerksam zu machen und ihn direkt zur Interaktion mit dem System zu leiten.

Das Design ist minimalistisch gehalten, um den Fokus auf die Navigation zu legen. Die Navigation selbst befindet sich in der gemeinsamen Layout-Komponente (+layout.svelte) und enthält alle relevanten Menüpunkte: Home, Books, Readlist und Add Book.

**Dateien:**

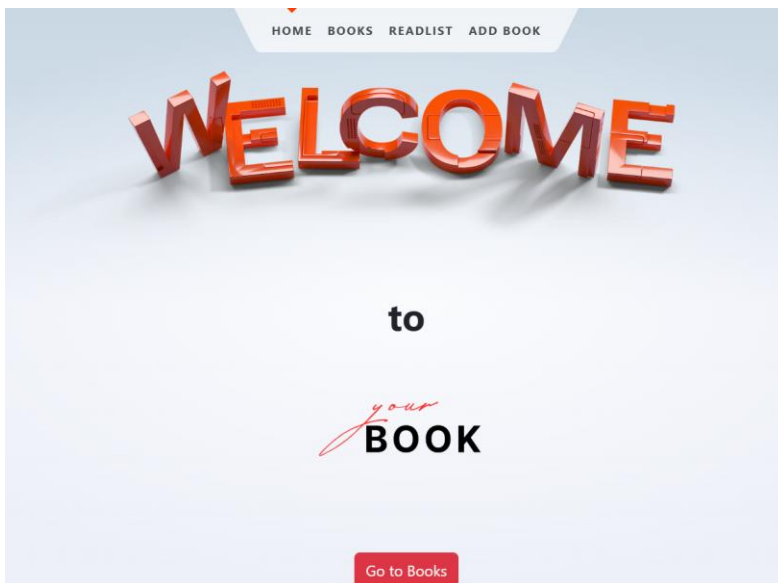
routes/+page.svelte

routes/+page.js

routes/+layout.svelte

routes/Header.svelte

lib/images



#### 3.2. Bücherübersicht

**Route: /books**

Auf dieser Seite werden alle Bücher angezeigt, die in der MongoDB gespeichert sind. Die Daten werden serverseitig geladen, inklusive des zugehörigen Autors über author\_id abgerufen. Jedes Buch wird mit Titel, Genre, Preis, Autorname und einem Bild dargestellt.

Zudem stehen zwei Buttons zur Verfügung:

Buy: führt zur Detailansicht des Buchs

Readlist: fügt das Buch zur persönlichen Merkliste (Readlist) hinzu

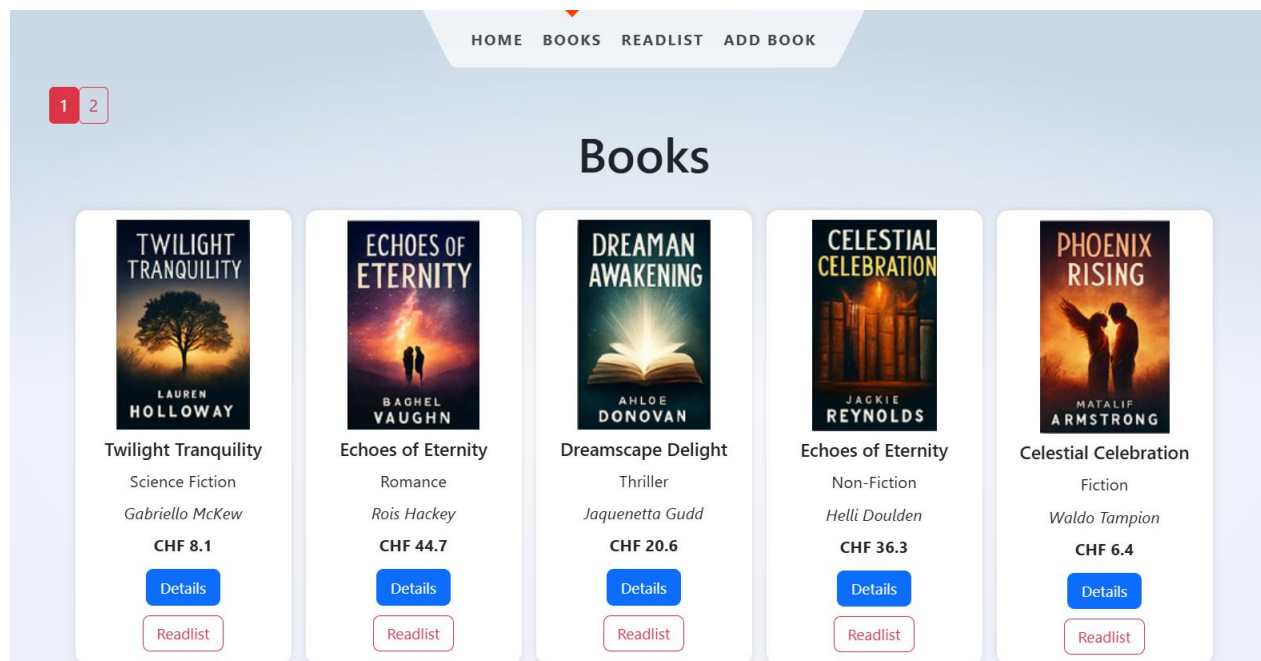
Die Buchbilder werden aus dem Verzeichnis `/static/covers` geladen. Für Bücher ohne definiertes Bild wird automatisch ein Platzhalter angezeigt.

#### Dateien:

`routes/books/+page.svelte`

`routes/books/+page.server.js`

`static/covers`



### 3.3. Detailansicht mit Details-Button

**Route:** `/books/[id]`

Wenn man auf „Details“ klickt, wird die Detailansicht des Buchs geöffnet. Hier werden die vollständigen Informationen des Buches angezeigt, basierend auf der `book_id`. Die Daten werden serverseitig direkt aus MongoDB geladen. Ist ein Buch nicht vorhanden oder die ID ungültig, wird ein entsprechender Fehlerstatus zurückgegeben.

#### Dateien:

`routes/books/[id]/+page.svelte`

`routes/books/[id]/+page.server.js`

### 3.4. Readlist

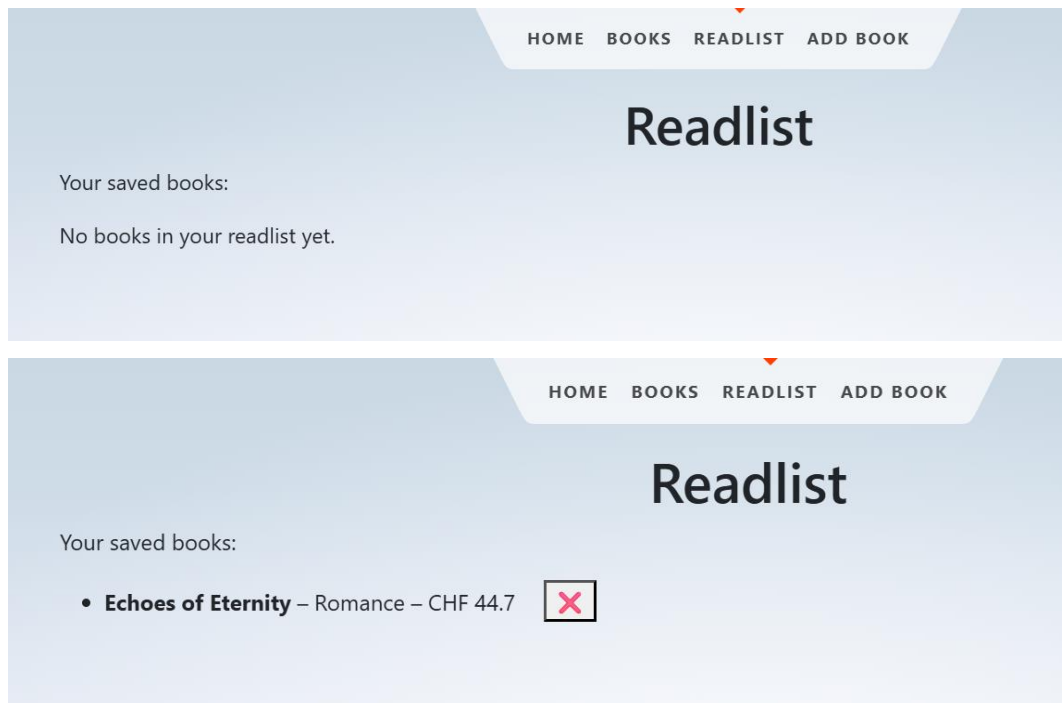
**Route:** `/readlist`

Diese Seite zeigt alle Bücher an, die der Benutzer zur Readlist hinzugefügt hat. Die Speicherung erfolgt lokal im Browser über `localStorage`.

Auf der Seite können die Bücher nicht nur angezeigt, sondern auch wieder entfernt werden. Neben jedem Eintrag gibt es einen „Remove“-Button, der das entsprechende Buch aus der Liste löscht. Das Entfernen aktualisiert sofort den localStorage und die Anzeige.

**Dateien:**

routes/readlist/+page.svelte



### 3.5. Add book

**Route:**/addbook

Auf „Add Book“ kann der Benutzer manuell ein neues Buch hinzufügen. Die Felder Titel, Genre und Preis werden eingegeben und das Buch wird sofort zur readlist hinzugefügt. Danach erfolgt eine automatische Weiterleitung zur Readlist-Seite. Die Bücher aus dieser Eingabe werden ebenfalls im localStorage gespeichert.

**Dateien:**

routes/addbook/+page.svelte

The image shows a screenshot of the 'Add a New Book' form. The navigation bar at the top has links: HOME, BOOKS, READLIST, and ADD BOOK (active). The title 'Add a New Book' is centered, with the subtitle 'to your readlist:' below it. The form consists of three input fields: 'Title', 'Genre', and 'Price (CHF)'. Below these fields is a blue button labeled 'Add Book'.

## 4. Erweiterungen

### 4.1. Lokale Speicherung der Readlist mit localStorage

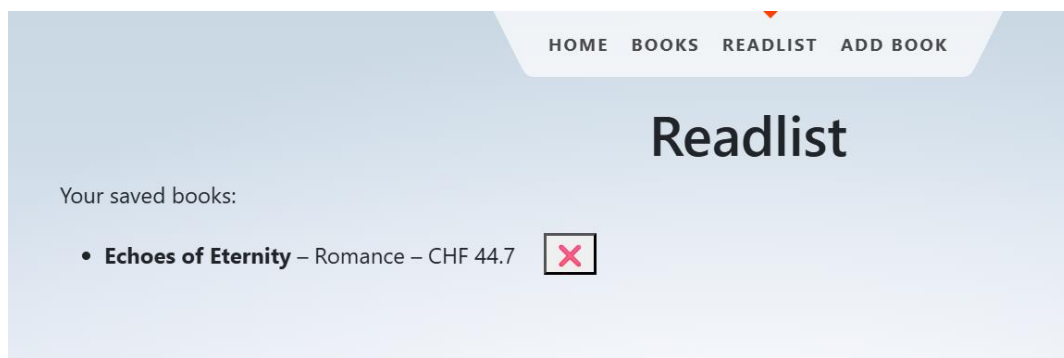
Die Bücher, die zur Readlist hinzugefügt werden, werden nicht in der Datenbank, sondern lokal im Browser über localStorage gespeichert. Dadurch bleiben sie auch nach einem Neuladen der Seite erhalten, ohne dass ein Benutzerkonto nötig ist.

**Dateien:**

routes/books/+page.svelte

routes/readlist/+page.svelte

routes/addbook/+page.svelte



### 4.2. Automatische Weiterleitung nach Hinzufügen

Nach dem Hinzufügen eines Buchs über „Readlist“-Button oder „Add Book“ erfolgt eine automatische Weiterleitung zur Readlist-Seite, um sofort Feedback zu geben.

**Dateien:**

routes/books/+page.svelte

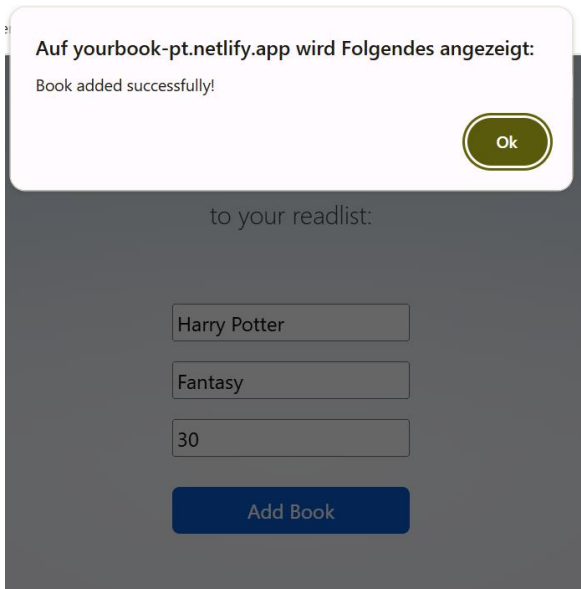
routes/addbook/+page.svelte

### 4.3. Benutzerfeedback durch Alert-Fenster

Beim erfolgreichen Hinzufügen eines neuen Buches über die Seite /addbook wird eine Meldung "Book added successfully!" angezeigt. Dadurch erhält der Nutzer ein direktes Feedback, dass die Aktion erfolgreich war.

**Datei:**

routes/addbook/+page.svelte

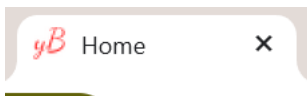


#### 4.4. Eigenes Favicon integriert

Die Anwendung verwendet ein eigenes Favicon, das im static-Verzeichnis abgelegt ist. Dadurch erscheint im Browser-Tab nicht das Standard-Icon, sondern ein individuelles Symbol passend zum Projekt „yourBook“. Dies unterstützt das Wiedererkennen der Anwendung und sorgt für ein professionelles Erscheinungsbild.

##### Datei:

static/favicon.png



#### 4.5. Paginierung der Buchübersicht

Die Buchübersicht auf der Seite /books wurde in zwei Seiten unterteilt, um die Darstellung übersichtlich zu halten. Auf der ersten Seite werden die ersten 15 Bücher angezeigt, auf der zweiten die restlichen.

Die Navigation erfolgt über nummerierte Seiten-Buttons oberhalb der Bücherliste. So wird die Benutzeroberfläche bei einer grösseren Datenmenge klar strukturiert und ordentlich gehalten.

##### Dateien:

routes/books/+page.svelte

routes/books/page2/+page.svelte

routes/books/+page.server.js

routes/books/page2/+page.server.js

