Prototyping Projektdokumentation

Name: Lorenz Eberhard

E-Mail: eberhlor@students.zhaw.ch

URL der deployten Anwendung: <https://bibliothek-prototyping.netlify.app/>

Git: <https://github.com/eberhlor/finalesprojekt-prototyping-bibliothek-eberhlor>

# Einleitung

Meine App ist die Bibliothek-eberhlor APP. Hier werden verschiedene Buchempfehlungen erstellt und angezeigt. Die Webseite enthält eine Schnittstelle zu einer externen API, welche Zitate als Inspiration für den Nutzer lädt. Unter dem Reiter Books werden alle Bücher angezeigt und es können auch neue Bücher erstellt werden. Über die BookCard kann der User Detailinformationen zu einem Buch anschauen. Dafür werden über den Fremdschlüssel sowohl das Genre als auch der Autor des Buches geladen. Unter dem Reiter Genres werden alle Genres angezeigt. Mit einem Click auf den Button «Books» wird eine Abfrage getriggert, welche die «GenreId» benutzt, um über einen Lookup in der Tabelle Books alle zugehörigen Bücher anzuzeigen. Unter dem Reiter «Authors» werden die Autoren angezeigt, über die Filterfunktion können Autoren anhand ihres Geschlechts auf der Seite angezeigt werden. Ausserdem kann man über den Button Books alle Bücher anschauen, welche von dem ausgewählten Autor geschrieben wurden. Es gibt zusätzlich eine Funktion, um neue Autoren zu erstellen.

# Datenmodell

Ein Bild, das Diagramm, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.

# Beschreibung der Anwendung

## Startseite

Route: /

Auf der Startseite ist ein YouTube Video als Iframe eingebunden. Darüber ist die Erweiterung der Zitate API zu finden.

Hier wird per get ein http Request an die API geschickt. Diese wird von einem API-Key geschützt. Hier könnte es auch andere Mechanismen geben, um den Endpunkt zu schützen. Weitere Details siehe Kapitel Einbindung externe API.

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

KI-generierte Inhalte können fehlerhaft sein.

Dateien:

* Routes/+page.svelte
* Routes/+page.server.js

## Bücherübersicht

Route: /books

Auf dieser Seite werden alle Bücher angezeigt. Für die Darstellung wird die Komponente BookCard.svelte verwendet.

Mit einem Click auf die Bookcard wird die Detailpage des Buches geladen books/[id]. Über den Button Add Book wird man auf die Seite books/create weitergeleitet.

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

KI-generierte Inhalte können fehlerhaft sein.

Dateien:

* Routes/books/+page.svelte
* Routes/books/+page.server.js
* lib/components/BookCard.svelte

## Bücherdetail

Route: /books/[id]

Auf dieser Seite werden alle Detailinformationen zu einem Buchgeladen. Zusätzlich wird hier auch der Name des Autors und das Genre geladen und über die Komponente BookdetailCard.svelte dargestellt. Genauere Informationen dazu siehe Kapitel Erweiterungen. Über den Back Button wird man zurück auf die Seite /books weitergeleitet.

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

KI-generierte Inhalte können fehlerhaft sein.

Dateien:

* Routes/books/[book\_id]/+page.svelte
* Routes/books/[book\_id]/+page.server.js
* lib/components/BookdetailCard.svelte

## Create Bücher Form

Route: /books/create

Auf dieser Seite kann ein neues Buch erstellt werden. Das Form wurde noch erweitert, dass man bei der Erstellung eines neuen Buches auch die bestehenden Genres und Autoren aus dem Dropdown laden kann und dem Buch hinzufügen kann. Genauere Informationen werden in den Erweiterungen beschrieben. Über den

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

KI-generierte Inhalte können fehlerhaft sein.

Dateien:

* Routes/books/create/+page.svelte
* Routes/books/create/+page.server.js

## Genres

Route: /genres

Auf dieser Seite findet man eine Übersicht aller Genres. Die Genre Information werden über die GenreCard.svelte dargestellt. Mit einem Click auf den Button Books werden die Daten zu dem jeweiligen Genre angezeigt /genres/[id].

Ein Bild, das Text, Screenshot, Grafikdesign, Tanz enthält.

KI-generierte Inhalte können fehlerhaft sein.

Dateien:

* Routes/genres/+page.svelte
* Routes/genres/+page.server.js
* lib/components/GenreCard.svelte

## Bücher von Genre anzeigen

Route: /genres/[id]

Auf dieser Seite werden alle Bücher mit demselben Genre angezeigt. Die Daten werden wieder über die BookCard.svelte Komponente angezeigt.

Ein Bild, das Text, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

Dateien:

* Routes/genres/[genre\_id]/+page.svelte
* Routes/genres/[genre\_id]/+page.server.js
* lib/components/BookCard.svelte

## Autorenübersicht

Route: /authors

Auf dieser Seite werden die Buchautoren angezeigt. Über den Button Add new Author kann ein neuer Autor erstellt werden /authors/create. Im Dropdown kann die Anzeige der Autoren noch angepasst werden, in dem man nach dem Geschlecht der Autoren filtert.

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

KI-generierte Inhalte können fehlerhaft sein.

Dateien:

* Routes/authors/+page.svelte
* Routes/authors/+page.server.js

## Autordetail Page (Übersicht von allen Büchern die von einem Autor geschrieben wurden)

Route: /authors/[author\_id]

Wenn man auf den Books Button drückt, wird man auf diese Seite weitergleitet. Auf dieser Seite werden alle Bücher angezeigt, welche von dem Autor geschrieben wurden. Die Daten werden über die BookCard Komponente angezeigt. Weiter Informationen sind im Kapitel der Erweiterungen beschrieben.

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

KI-generierte Inhalte können fehlerhaft sein.

Dateien:

* Routes/authors/[author\_id]/+page.svelte
* Routes/authors/[author\_id]/+page.server.js

## Create Autor Form

Route: /authors/create

Das Create Autorform ist ein simples Formular, welches einen neuen Autor lässt. Danach hat man die Möglichkeit im Create Buchform ein neues Werk hinzuzufügen. Über den Back Button kehrt der Benutzer auf die /authors Page zurück. Für die Formatierung des Geburtsdatums wird derselbe Code benutzt, wie in den Erweiterungen beschrieben.

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

KI-generierte Inhalte können fehlerhaft sein.

Dateien:

* Routes/authors/create/+page.svelte
* Routes/authors/create/+page.server.js

# Erweiterungen

## Autor-Gender-Filter

Auf der Seite /authors können via Dropdown Autoren ausgeblendet werden, die nicht dem Filterkriterium entsprechen. Die Logik wurde ausschliesslich im html umgesetzt. Dabei wird über ein If geprüft, ob der Inhalt des selectedGenders ein leerer String ist. In diesem Fall würden alle Autoren angezeigt werden. Ansonsten werden alle Autoren angezeigt, welche dem selectedGender entsprechen. Um die Komplexität etwas zu vereinfachen, wurde bewusst darauf verzichtet das Dropdown mit den Unique Genders aus der Datenbank zu befüllen. Da in der aktuellen Lösung nur zwei Genders hinterlegt sind «Female/ Male» wurde das Dropdown fix mit diesen Wertenbefüllt.

Datei:

* routes/authors/+page.svelte

## Footer

Ich habe im Layout einen HTML Footer hinzugefügt. Dieser wird auf jeder Seite am Ende der Page angezeigt.

Datei:

* Routes/layout.svelte

## Einbindung von externer API

Ich habe eine externe API eingebunden, über diese API werden Zitate geladen, welche auf der Startseite angezeigt werden.Die Zitate werden über ein get im Json Format geladen. Für die Authentifizierung in der API wird ein API-Key mitgeschickt, welcher von mir nach dem Anmelden generiert wurde.

Ein Bild, das Text, Screenshot, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Dateien:

* Routes/+page.svelte
* Routes/+server.svelte

## Erweiterung von Book Forms

Durch mein Datenmodell kann ich in meiner Bücher Entität auch einen Autor und ein Genre speichern. Deshalb wird in meinem Post (Create) Book Forms auch jeweils ein Dropdown zur Verfügung gestellt, welches einer Liste aller bestehenden Autoren und Genres enthält.

Dateien:

* Routes/books/create/+page.svelte
* Routes/books/create/+page.svelte
* Lib/db.js

## Detail Bücherpage

Für meine Buchdetailseite musste ich eine komplexere Abfrage mit lookup schreiben und mit unwind aus dem Array ein Objekt für den Autor und das Genre erstellen. Als Result erhalte ich die Bücherdaten und die dazugehörigen Daten Autor und Genre Objekte.

Dateien:

* Routes/books/[book\_id]/+page.svelte
* Routes/books/[book\_id]/+server.svelte
* Lib/db.js

## Bücher pro Genre anzeigen

Für meine Genre Detailpage habe ich eine Abfrage geschrieben, welche alle Bücher zu dem ausgewählten Genre anzeigt. Die komplexere Abfrage wird über ein Lookup gemacht, das Array wird mit einem Unwind in einzelne Objekte aufgeteilt. Mit dem Project Befehl konnte ich Daten filtern und spezifisch auf meine Bedürfnisse zurückgeben.

Dateien:

* Routes/genres/[genre\_id]/+page.svelte
* Routes/genres/[genre\_id]/+server.svelte
* Lib/db.js

## Bücher pro Author anzeigen

Für meine Author Detailpage habe ich eine Abfrage geschrieben, welche alle Bücher zu dem ausgewählten Author anzeigt. Die komplexere Abfrage wird über ein Lookup gemacht, das Array wird mit einem Unwind in einzelne Objekte aufgeteilt. Mit dem Project Befehl konnte ich Daten filtern und spezifisch auf meine Bedürfnisse zurückgeben.

Dateien:

* Routes/authors/[author\_id]/+page.svelte
* Routes/authors/[author\_id]/+server.svelte
* Lib/db.js

## Datenbank

Ich habe mein Datenmodell so aufgebaut, dass die Books Collection einen Fremdschlüssel für den Autor und das Genre enthält. Damit kann ich über den Fremdschlüssel sowohl das Genre als auch der Autor des Buches geladen.

Ein Bild, das Text, Schrift, Reihe, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Text, Schrift, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

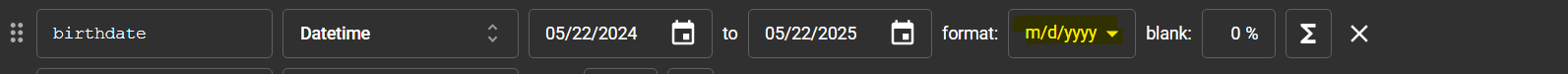
Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

## Formatierung von Geburtsdatum

Da bei der Generierung der Testdaten das Geburtsdatum im Format Monat/ Tag/ Jahr generiert wurde, musste ich für die Darstellung im Html das Datum noch neu formatieren. Dafür habe ich im Skript teil eine Funktion geschrieben, die als Parameter das Geburtsdatum erhält. Zuerst wird aus dem String ein neues Datum erstellt. Aus diesem Datum wird der Tag, Monat und das Jahr extrahiert. Bei den Tagen und Monaten wird geprüft, ob der String zwei stellen hat, 07 oder 7, wenn 7 stehen würde, wird vor eine 0 davor geschrieben. 07 Januar statt 7 Januar. Bei der Extrahierung vom Datum muss für die Darstellung + 1 gerechnet werden, da die Monate im Date von 0-11 dargestellt werden.

Konvertierungsbeispiel: 6/24/1990=> Sun Jun 24 1990 00:00:00 GMT+0200 (Mitteleuropäische Sommerzeit)



Datei:

* routes/authors/+page.svelte

## Eibindung von Lucide Library

Ich habe für die Darstellung der Daten in der Bücherdetail Seite und auf weiteren Seiten die open-source Icons von Luicide eingebunden.

Ein Bild, das Text, Buch, Poster, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Dateien:

* Routes/books/[book\_id]/+page.svelte
* Routes/authors/+page.svelte
* Routes/contact/+page.svelte
* Package.json