Prototyping Projektdokumentation

Name: Diaz Murga Cesar Michael

E-Mail: diazmces@students.zhaw.ch

URL der deployten Anwendung: <https://buecher-sveltekit-jcbkesayt-mikes-projects-f363e9bf.vercel.app/genres/683bfc7f9f0c4567a5604038/books>  
URL Github-Repository: <https://github.com/MikeDiazxyz/buecher-sveltekit>

# Einleitung

Die Bücherverwaltung ist eine Webanwendung zur übersichtlichen Organisation von Büchern und Genres. Nutzer können neue Bücher und Genres erfassen, bestehende bearbeiten und löschen sowie eine Such- und Sortierfunktion verwenden. Die Anwendung ist responsiv und passt sich mobilen Geräten an.

# Datenmodell

# Ein Bild, das Entwurf, Diagramm, Zeichnung, Kreis enthält. KI-generierte Inhalte können fehlerhaft sein.

# Beschreibung der Anwendung

## Startseite

**Ein Bild, das Text, Screenshot, Rechteck, Bilderrahmen enthält.

KI-generierte Inhalte können fehlerhaft sein.Route:** / (Root)

Die Startseite begrüßt die Nutzer:innen mit einem Willkommensbild, einer kurzen Beschreibung der Anwendung und Hinweisen zur Nutzung. Sie dient als freundlicher Einstiegspunkt in die Bücherverwaltung und verweist über die Navigation auf die Bücher- und Genreübersichten.

**Dateien:**

* src/routes/+page.svelte (Layout und Text der Startseite)
* Optional: src/routes/+page.server.js (Falls serverseitige Logik benötigt wird, z.B. Daten laden)

## Bücherübersicht

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

KI-generierte Inhalte können fehlerhaft sein.

**Route:** /books

**Beschreibung:**

Zeigt eine Liste aller Bücher in einer sortier- und filterbaren Tabelle. Nutzer können Bücher suchen, sortieren, bearbeiten oder löschen. Außerdem gibt es einen Button zum Erfassen eines neuen Buchs.

* Suchfeld filtert dynamisch nach Titel und Autor.
* Sortierung nach Titel, Autor und Erscheinungsjahr (auf- und absteigend).
* Bearbeiten-Button führt zur Detailseite mit Formular zur Buchänderung.
* Löschen-Button entfernt Buch mit Bestätigung, ohne die Seite neu zu laden.
* Button zum Erfassen eines neuen Buchs führt zur Bucherfassungsseite.

**Wichtige Dateien:**

* src/routes/books/+page.svelte (Liste mit Such-, Sortier- und Aktionsfunktionen)
* src/routes/books/+page.server.js (Lädt Bücher, verarbeitet Lösch-Requests)
* src/lib/index.js (CRUD-Funktionen: getAllBooks, deleteBook)

**Workflow:**

1. Nutzer öffnet /books.
2. Tabelle zeigt alle Bücher.
3. Nutzer filtert, sortiert oder navigiert.
4. Klick auf „✏️ Bearbeiten“ öffnet Buchbearbeitung.
5. Klick auf „🗑️ Löschen“ entfernt Buch nach Bestätigung.
6. Klick auf „➕ Neues Buch erfassen“ öffnet Eingabeformular.

## Buch erstellen

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

KI-generierte Inhalte können fehlerhaft sein.

**Route:** /books/create

**Beschreibung:**

Formularseite zum Hinzufügen eines neuen Buchs. Nutzer geben Titel, Autor, Jahr und Genre ein. Nach dem Absenden wird das Buch in der Datenbank gespeichert und Nutzer zurück zur Bücherübersicht geleitet.

* Eingabefelder für Titel, Autor, Erscheinungsjahr und Genre (Dropdown mit allen Genres).
* Pflichtfelder validieren Eingaben.
* Nach erfolgreichem Speichern Weiterleitung zu /books.
* Erfolgs- und Fehlermeldungen werden angezeigt.

**Wichtige Dateien:**

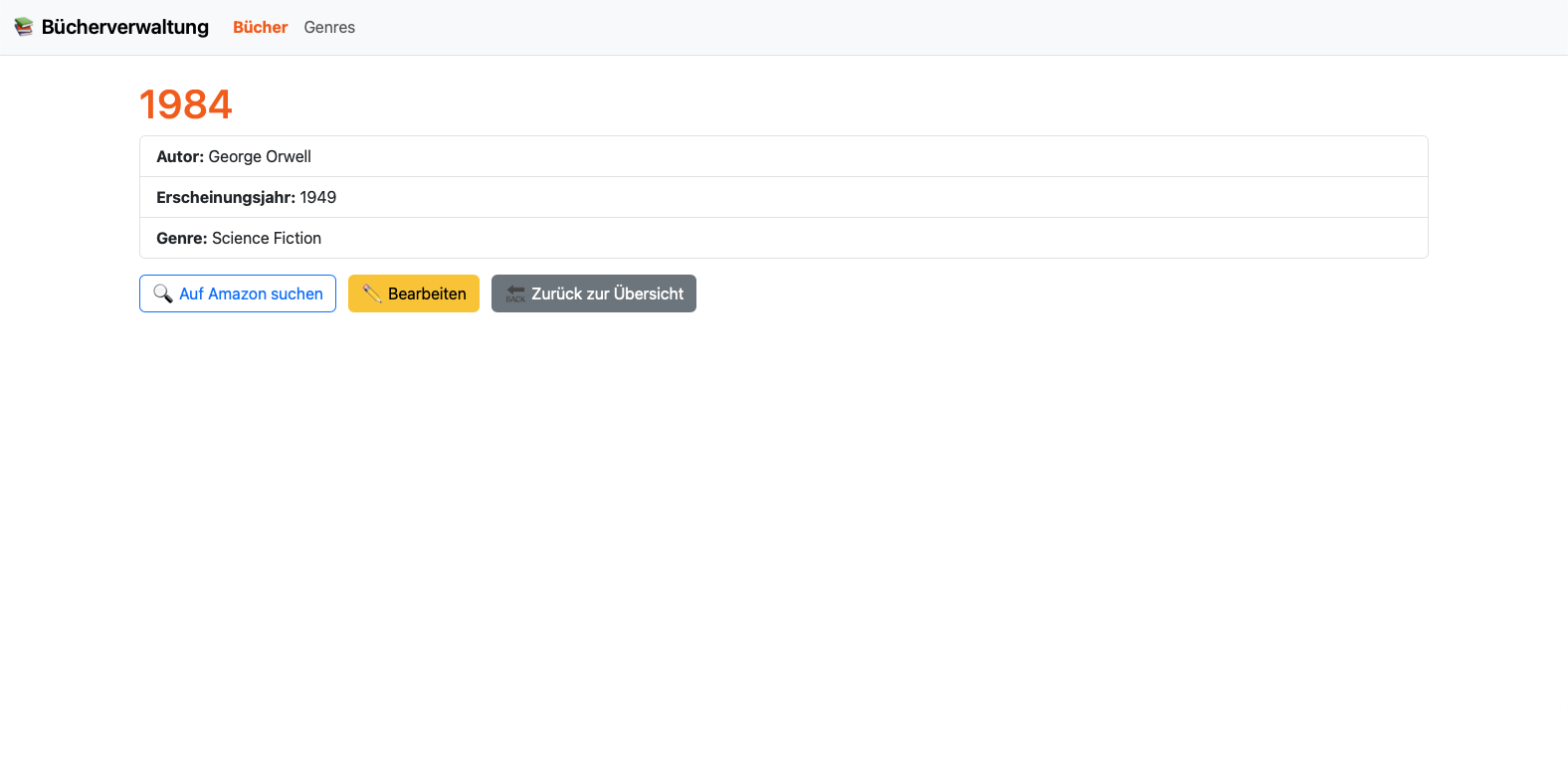
* src/routes/books/create/+page.svelte (Formular UI)
* src/routes/books/create/+page.server.js (Formularverarbeitung und Buchspeicherung)
* src/lib/index.js (CRUD-Funktion: addBook)

**Workflow:**

1. Nutzer öffnet /books/create.
2. Füllt Formular aus und sendet ab.
3. Server validiert und speichert Buch.
4. Weiterleitung zur Bücherübersicht.

## Buch bearbeiten

**Route:** /books/[book\_id]



**Beschreibung:**

Formularseite zur Bearbeitung eines bestehenden Buchs. Felder sind mit den aktuellen Buchdaten vorausgefüllt. Nach Änderung werden die Daten gespeichert und Nutzer zurück zur Bücherübersicht geleitet.

* Vorausgefüllte Felder für Titel, Autor, Jahr und Genre (Dropdown).
* Änderung wird validiert und gespeichert.
* Erfolgs- und Fehlermeldungen werden angezeigt.
* Löschfunktion mit Bestätigung möglich.
* Weiterleitung auf Amazon mit Suchbegriff (Titel + Autor)

**Wichtige Dateien:**

* src/routes/books/[book\_id]/+page.svelte (Formular mit vorausgefüllten Daten)
* src/routes/books/[book\_id]/+page.server.js (Lädt Buch, verarbeitet Updates und Löschungen)
* src/lib/index.js (CRUD-Funktionen: getBookById, updateBook, deleteBook)

**Workflow:**

1. Nutzer öffnet /books/[book\_id].
2. Sieht vorausgefülltes Formular.
3. Ändert Daten und speichert oder löscht Buch.
4. Nach Speichern oder Löschen Rückkehr zur Übersicht.

## Genres Übersicht

**Route:** /genres

Ein Bild, das Text, Screenshot, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

**Beschreibung:**

Tabelle mit allen Genres, inklusive Such- und Sortierfunktion. Nutzer können Genres bearbeiten, löschen oder alle Bücher eines Genres anzeigen lassen. Button führt zur Genre-Erfassung.

* Suchfeld filtert Genres nach Name und Beschreibung.
* Sortierung nach Name und Beschreibung.
* Bearbeiten- und Löschen-Buttons bei jedem Genre.
* Link zu Seite mit allen Büchern des Genres.
* Button zum Erfassen eines neuen Genres.

**Wichtige Dateien:**

* src/routes/genres/+page.svelte (Genre-Tabelle mit Such- und Sortierfunktionen)
* src/routes/genres/+page.server.js (Lädt Genres und verarbeitet Lösch-Requests)
* src/lib/index.js (CRUD-Funktionen: getAllGenres, deleteGenre)

**Workflow:**

1. Nutzer öffnet /genres.
2. Sieht alle Genres, sucht oder sortiert.
3. Klickt auf „✏️ Bearbeiten“, „🗑️ Löschen“ oder „📚 Alle Bücher dieses Genres“.
4. Klick auf „📝 Neues Genre erfassen“ öffnet Erfassungsformular.

## Genre erstellen und bearbeiten

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

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Text, Schrift, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

**Route:** /genres/create und /genres/[genre\_id]

**Beschreibung:**

Formulare zur Eingabe oder Bearbeitung von Genres. Nach Speichern wird zurück zur Genreübersicht geleitet.

* Eingabefelder für Name und Beschreibung.
* Validierung der Eingaben.
* Nach Speichern Weiterleitung zur Genreübersicht.
* Löschfunktion im Bearbeitungsformular möglich.

**Wichtige Dateien:**

* src/routes/genres/create/+page.svelte und src/routes/genres/[genre\_id]/+page.svelte (Formulare)
* src/routes/genres/create/+page.server.js und src/routes/genres/[genre\_id]/+page.server.js (Formularverarbeitung)
* src/lib/index.js (CRUD-Funktionen: addGenre, updateGenre, getGenreById)

**Workflow:**

1. Nutzer öffnet Formularseite zum Erstellen oder Bearbeiten.
2. Gibt Name und Beschreibung ein, speichert.
3. Rückkehr zur Genreübersicht.

# Erweiterungen

## Suchfunktion in Genre- und Bücherübersichten

Auf den Seiten /genres und /books wurde eine Suchfunktion implementiert, die das schnelle Finden von Einträgen durch Eingabe von Suchbegriffen ermöglicht. Die Filterung erfolgt clientseitig in den Komponenten +page.svelte mittels reactive statements ($: filteredBooks / $: filteredGenres).

Dateien:

* src/routes/books/+page.svelte
* src/routes/genres/+page.svelte

## Sortierfunktion in Genre- und Bücherübersichten

Die Tabellen in /genres und /books sind sortierbar nach verschiedenen Spalten (z. B. Titel, Autor, Name, Beschreibung). Das Umschalten der Sortierreihenfolge wurde in den Komponenten +page.svelte mit toggleSort() umgesetzt.

Dateien:

* src/routes/books/+page.svelte
* src/routes/genres/+page.svelte

## Dynamische Genre-Auswahl in Buchformularen

In den Bucherfassungs- und Bearbeitungsformularen (/books/create und /books/[book\_id]) wird die Liste der Genres dynamisch aus der Datenbank geladen und als Dropdown-Menü angezeigt, um eine konsistente Auswahl zu gewährleisten.

Dateien:

* src/routes/books/create/+page.svelte
* src/routes/books/create/+page.server.js
* src/routes/books/[book\_id]/+page.svelte
* src/routes/books/[book\_id]/+page.server.js

## Löschen ohne Seitenreload

Beim Löschen von Genres auf der Übersicht /genres wird das gelöschte Genre sofort aus der Liste entfernt, ohne die Seite neu zu laden. Dies wird mittels Fetch-Request und clientseitigem State-Update in +page.svelte umgesetzt.

Datei(en:

* src/routes/genres/+page.svelte
* src/routes/genres/+page.server.js

## Amazon-Suchlink in der Buch-Detailansicht

Auf der Buch-Detailseite /books/[book\_id] wird ein Link generiert, der eine Amazon-Suche mit Titel und Autor des Buchs startet. Somit kann der Nutzer das Buch direkt auf Amazon finden.

Dateien:

* src/routes/books/[book\_id]/+page.svelte