

# Prototyping Projektdokumentation

---

Name: Gabriel Ceccato

E-Mail: ceccagab@students.zhaw.ch

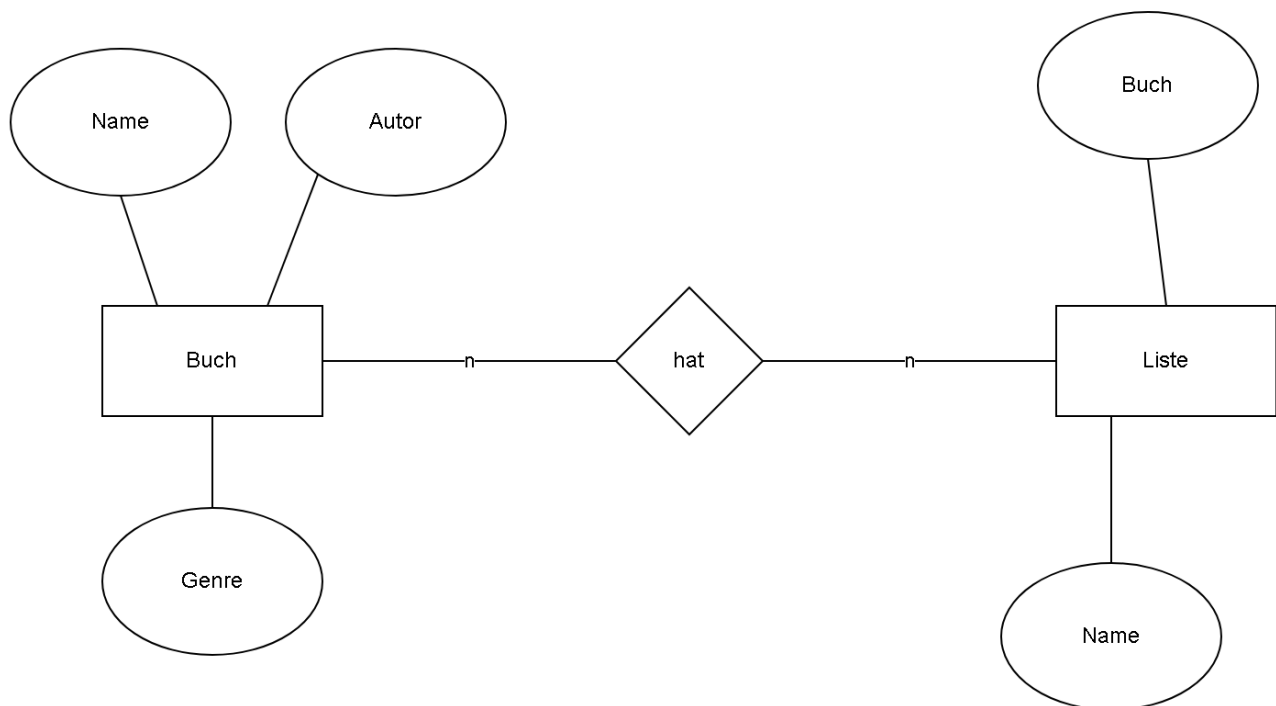
URL der deployten Anwendung: <https://bookspt.netlify.app/>

## 1. Einleitung

BookBlaze ist eine webbasierte Anwendung zur Verwaltung persönlicher Büchersammlungen und Leselisten. Nutzer können Bücher mit Titel, Autor und Genre erfassen, vorhandene Einträge bearbeiten oder löschen und so ihre Bibliothek flexibel organisieren. Darüber hinaus lassen sich aus den gespeicherten Büchern beliebige Listen zusammenstellen, die zur Übersicht angezeigt oder bei Bedarf gelöscht werden können.

Alle Daten werden in einer MongoDB-Datenbank abgelegt, während das Node.js-Backend über eine REST-API die CRUD-Operationen (Erstellen, Auslesen, Aktualisieren, Löschen) bereitstellt. Das Frontend basiert auf Svelte und sorgt für ein responsives Nutzererlebnis mit Echtzeit-Aktualisierungen, sodass Änderungen an Büchern oder Listen unmittelbar sichtbar werden.

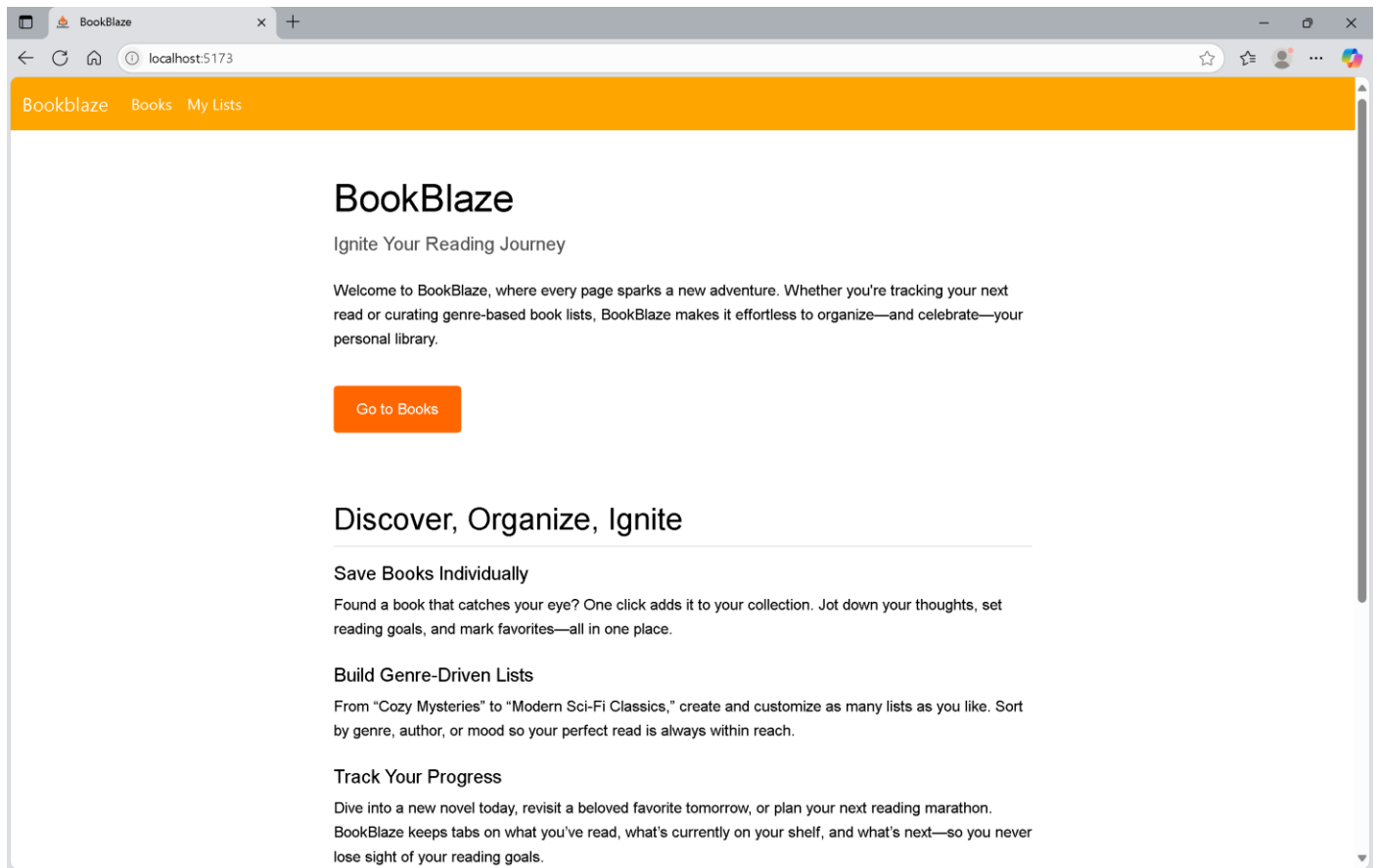
## 2. Datenmodell



## 3. Beschreibung der Anwendung

### 3.1. Homepage

Route: /routes



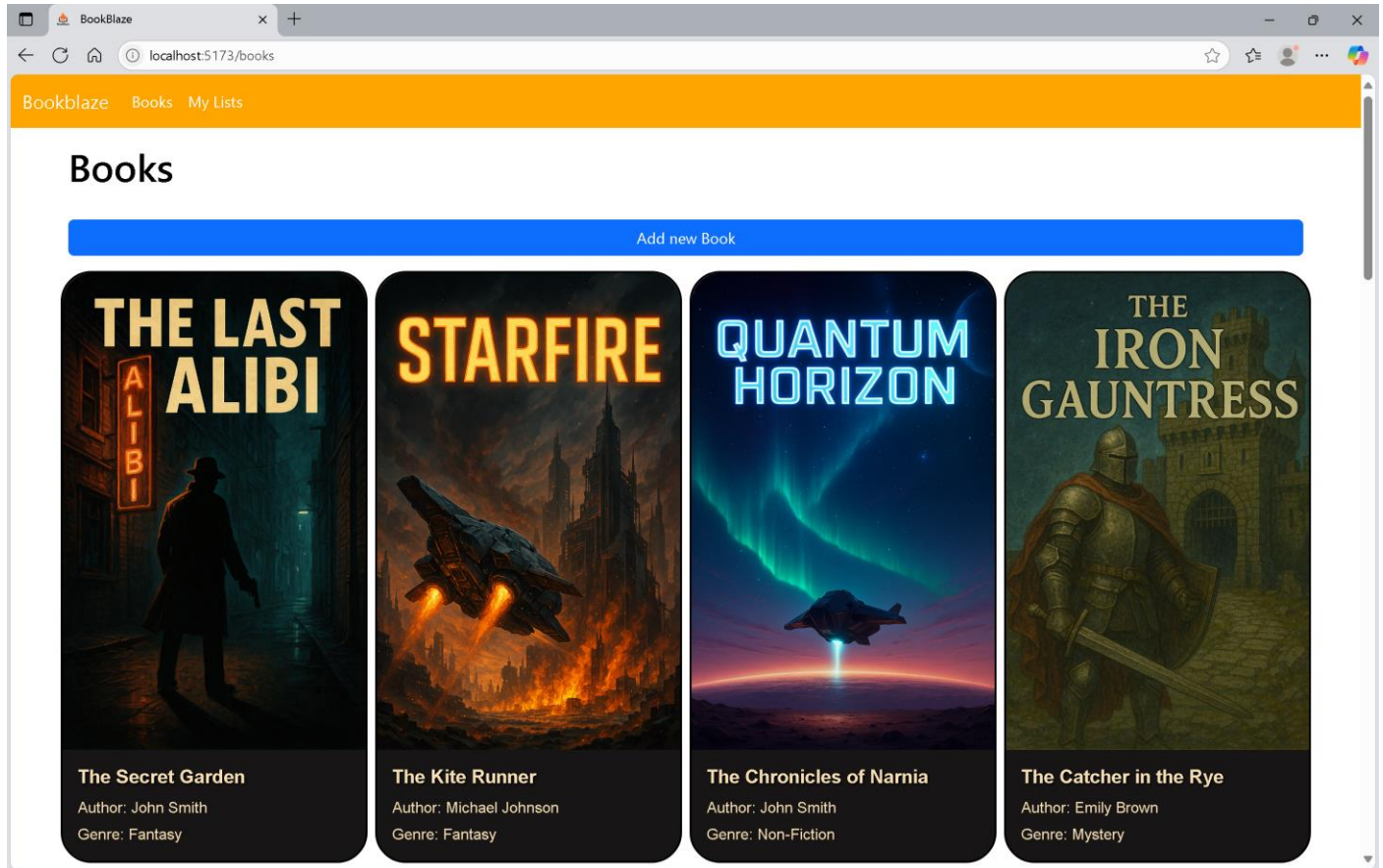
Die Startseite heißt neue oder wiederkehrende Nutzer willkommen, erklärt kurz den Sinn von BookBlaze und zeigt direkt die zentralen Funktionen. Über den Button «Go to books» gelangt man zur Seite /routes/books.

Dateien:

- routes/+page.svelte

### 3.2. Bücher-Übersicht (Books Page)

Route: /routes/books



Auf dieser Seite sieht der Nutzer seine gesamte Sammlung („Library“) in einer responsiven Grid-Ansicht. Jeder Eintrag ist ein sogenanntes BookCard-Component, das folgende Elemente enthält: Titel des Buchs, Autor, Genre, Mini-Coverbild (falls vorhanden) oder ein Platzhalter. Oben auf der Seite befindet sich eine Überschrift „Books“ und ein Button „Add New Book“, der zur Add-Book-Form (3.3) führt.

#### 1. Anzeige aller Bücher

- Beim Laden der Seite ruft das Svelte-Frontend per REST-API alle Buch-Datensätze aus der MongoDB ab.
- Diese Daten werden als Props an mehrere BookCard-Komponenten übergeben, die in einer flexiblen Grid-Struktur angeordnet sind.

#### 2. BookCard-Interaktionen

- Klickt man auf eine BookCard (z. B. das Cover oder den Titel), navigiert man zur Detail-View des jeweiligen Buches (3.4).
- Durch einen Hover erkennt man, dass die Karte klickbar ist.

#### 3. Neues Buch hinzufügen

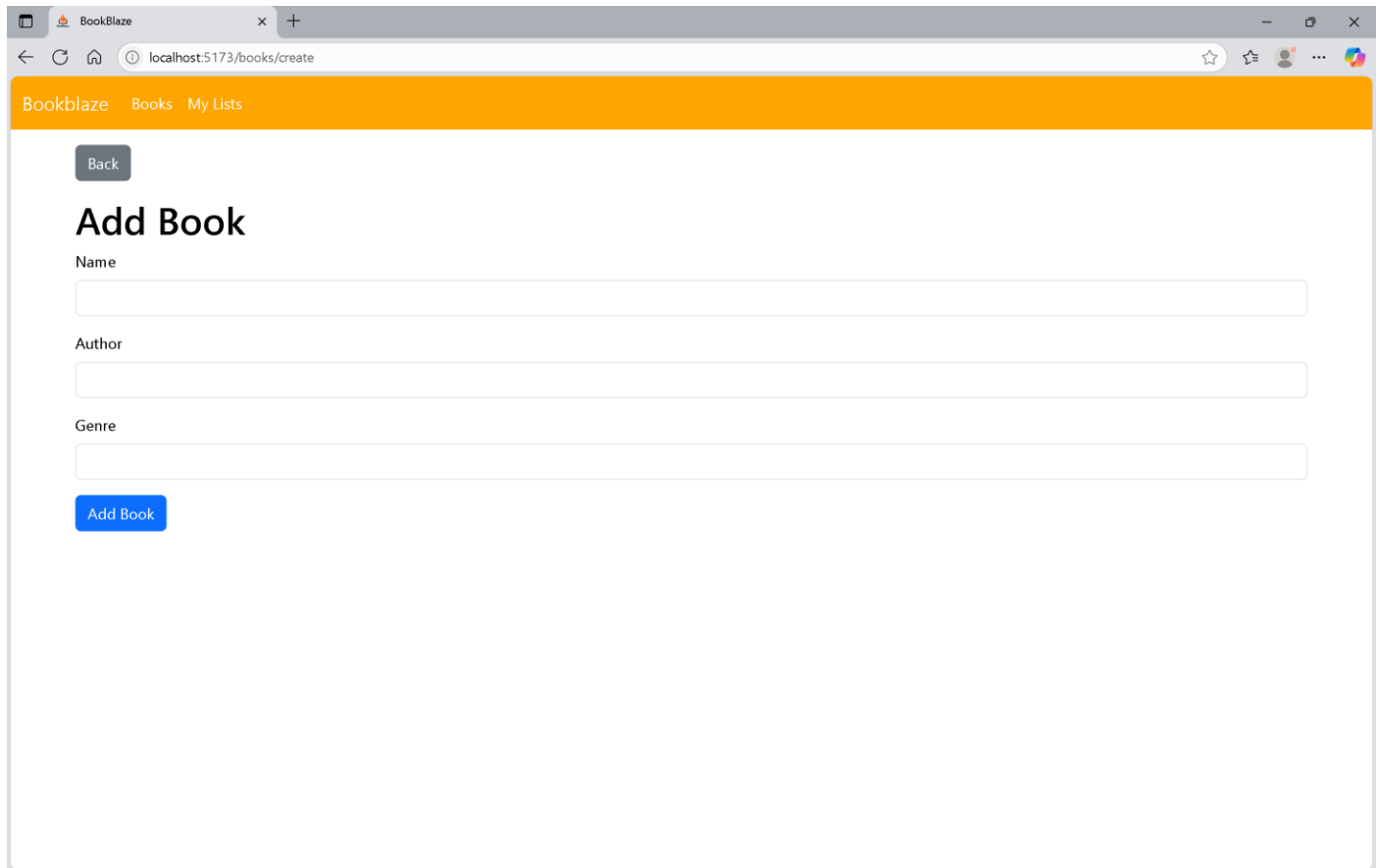
- Der „Add New Book“-Button führt per Link auf die Add-Book-Page (3.3).

Dateien:

- lib/components/BookCard.svelte
- routes/books/+page.svelte
- routes/books/+page.server.js

### 3.3. Buch-Formular (Add new Book)

Route: /books/create



The screenshot shows a web browser window with the address bar displaying 'localhost:5173/books/create'. The page has an orange header bar with the text 'Bookblaze Books My Lists'. Below the header, there is a 'Back' button in a grey box. The main heading is 'Add Book' in a large, bold font. Below the heading, there are three input fields labeled 'Name', 'Author', and 'Genre'. At the bottom of the form, there is a blue 'Add Book' button.

Hier kann der Nutzer ein neues Buch in die Datenbank eintragen. Die Seite ist ein zentriertes Formular mit folgenden Feldern:

- Book Name
- Author
- Genre
- Submit-Button: Beschriftet z. B. „Save Book“
- Back-Button mit Ziel /books (zurück zur Bücher-Übersicht)

Direkt unter dem Formular kann nach erfolgreichem Absenden ein Hinweis erscheinen:

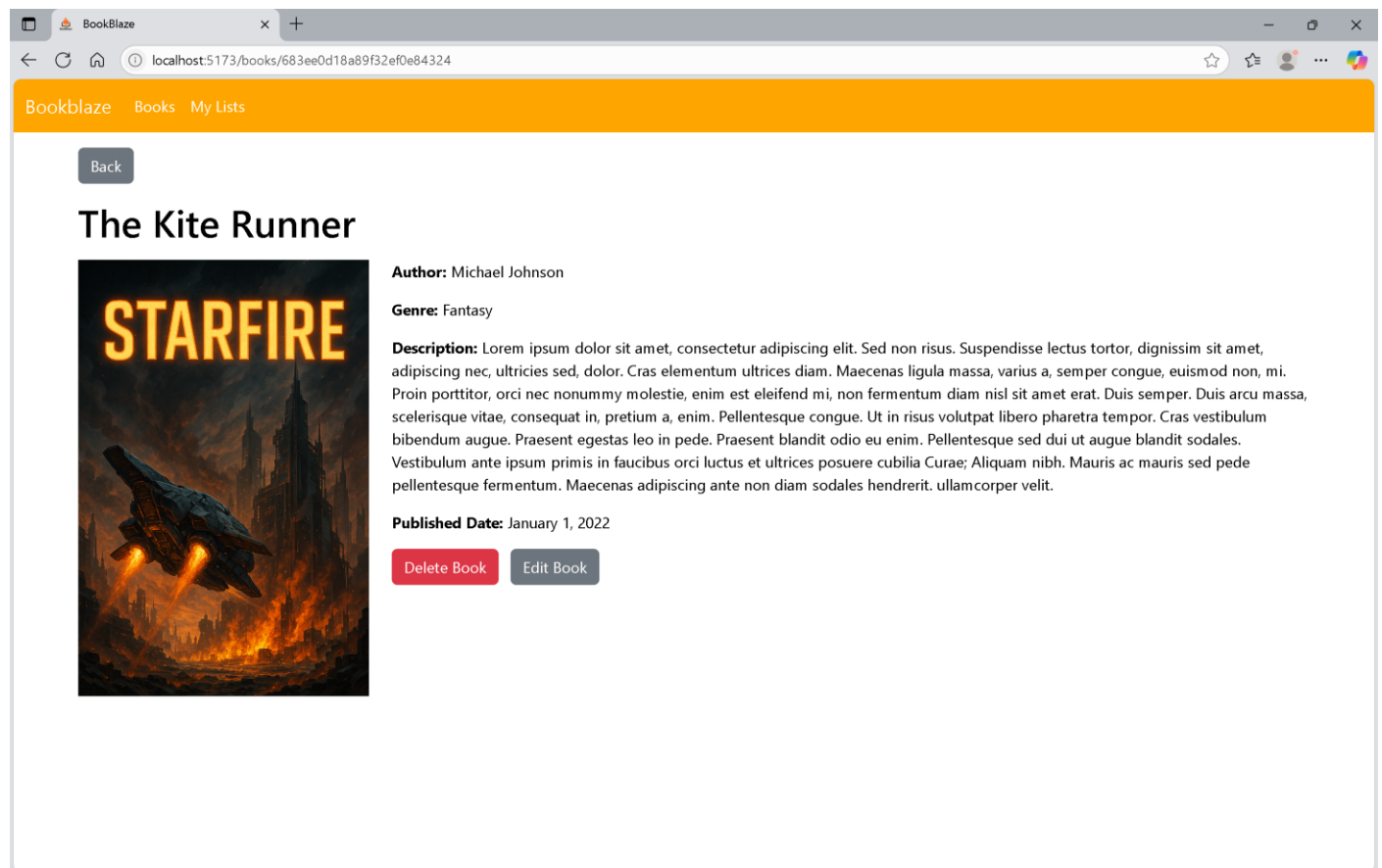
„Book saved.“

Dateien:

- lib/components/BookCard.svelte
- routes/books/create/+page.svelte
- routes/books/create/+page.server.js

### 3.4. Buch-Detailansicht

Route: /books/[book\_id]



Zeigt alle Informationen zu einem einzelnen Buch in übersichtlicher Detailansicht und hat einen «Delete Book» Button, der bei einem Klick das Buch aus MongoDB löscht und der Nutzer zurück auf die Bücher-Übersicht (3.2) geleitet. Der «Edit Book» Button ist ein Link, der zur Edit-Page (3.5) für dieses Buch navigiert (/books/:id/edit).

Dateien:

- lib/components/BookCard.svelte
- routes/books/[book\_id]/+page.svelte
- routes/books/book\_id]/+page.server.js

### 3.5. Buch bearbeiten (Edit Book)

Route: `/books/[book_id]/edit`

Bookblaze Books My Lists

## Edit Book

Book Name

Author

Genre

[Save Changes](#) [Cancel](#)

Book updated

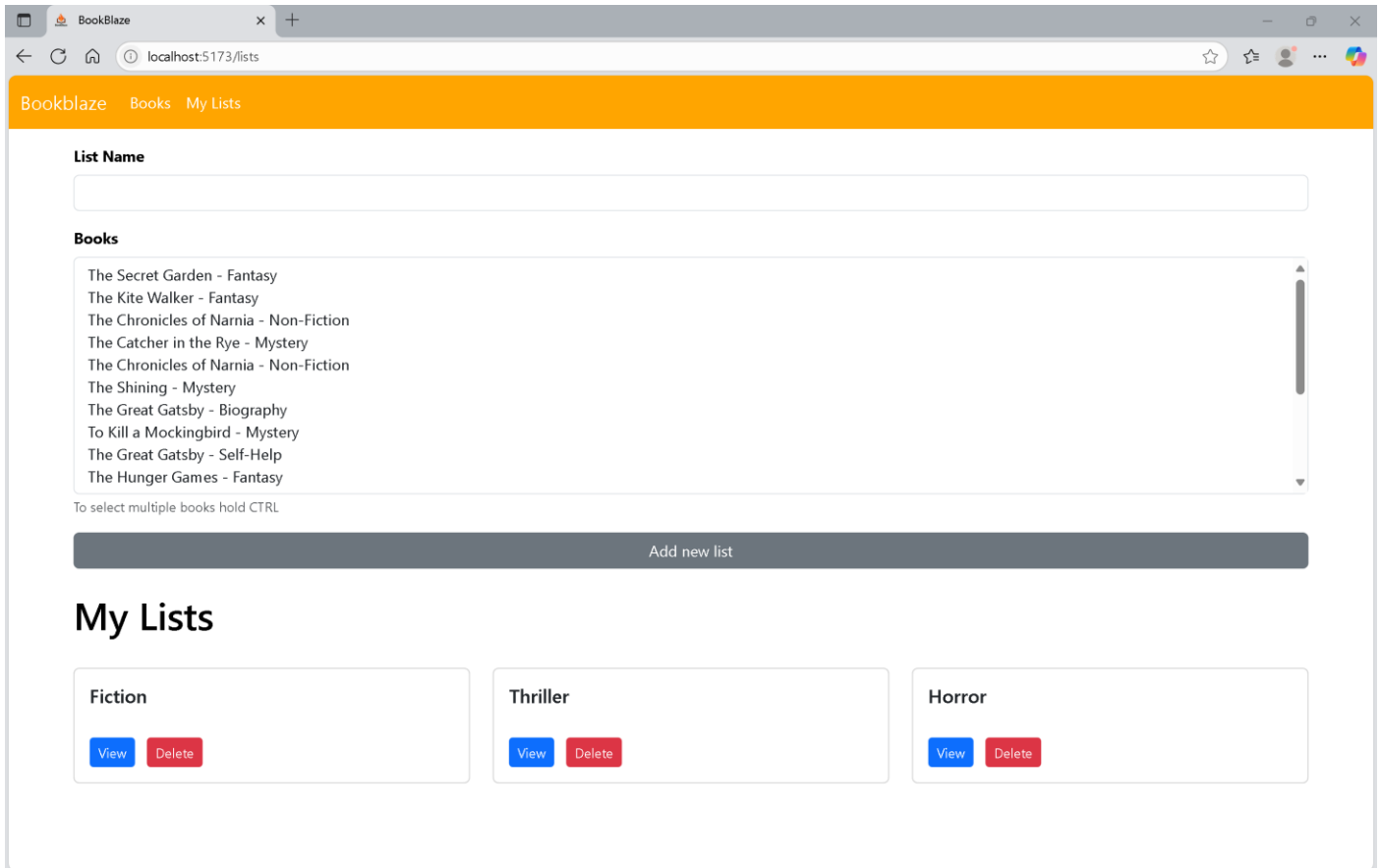
Hier kann der Nutzer die Metadaten eines vorhandenen Buchs ändern. Die Seite ähnelt dem Add-New-Book-Formular, ist jedoch mit bestehenden Werten vorbelegt. Durch Klick auf den «Save Changes» Button erscheint bei Erfolg die Meldung «Book updated». Mit dem «Cancel» Button geht man zurück zur Detailansicht `/books/[book_id]`.

Dateien:

- `lib/components/BookCard.svelte`
- `routes/books/book_id]/edit/+page.svelte`
- `routes/books/book_id]/edit/+page.server.js`

### 3.6. Listenverwaltung (Lists Page)

Route: `/routes/lists`



Auf der Listenverwaltungsseite kann der Nutzer neue Leselisten anlegen und bereits erstellte Listen einsehen oder löschen.

Zuoberst befindet sich ein Eingabebereich für eine neue Liste: Der Nutzer tippt einen Namen für die Liste ein und wählt per Mehrfachauswahlfeld (mit Hinweis „STRG gedrückt halten, um mehrere Bücher auszuwählen“) die gewünschten Bücher aus seiner Sammlung aus. Mit einem Klick auf „Create List“ wird die Liste angelegt und erscheint direkt unterhalb im Bereich der existierenden Listen.

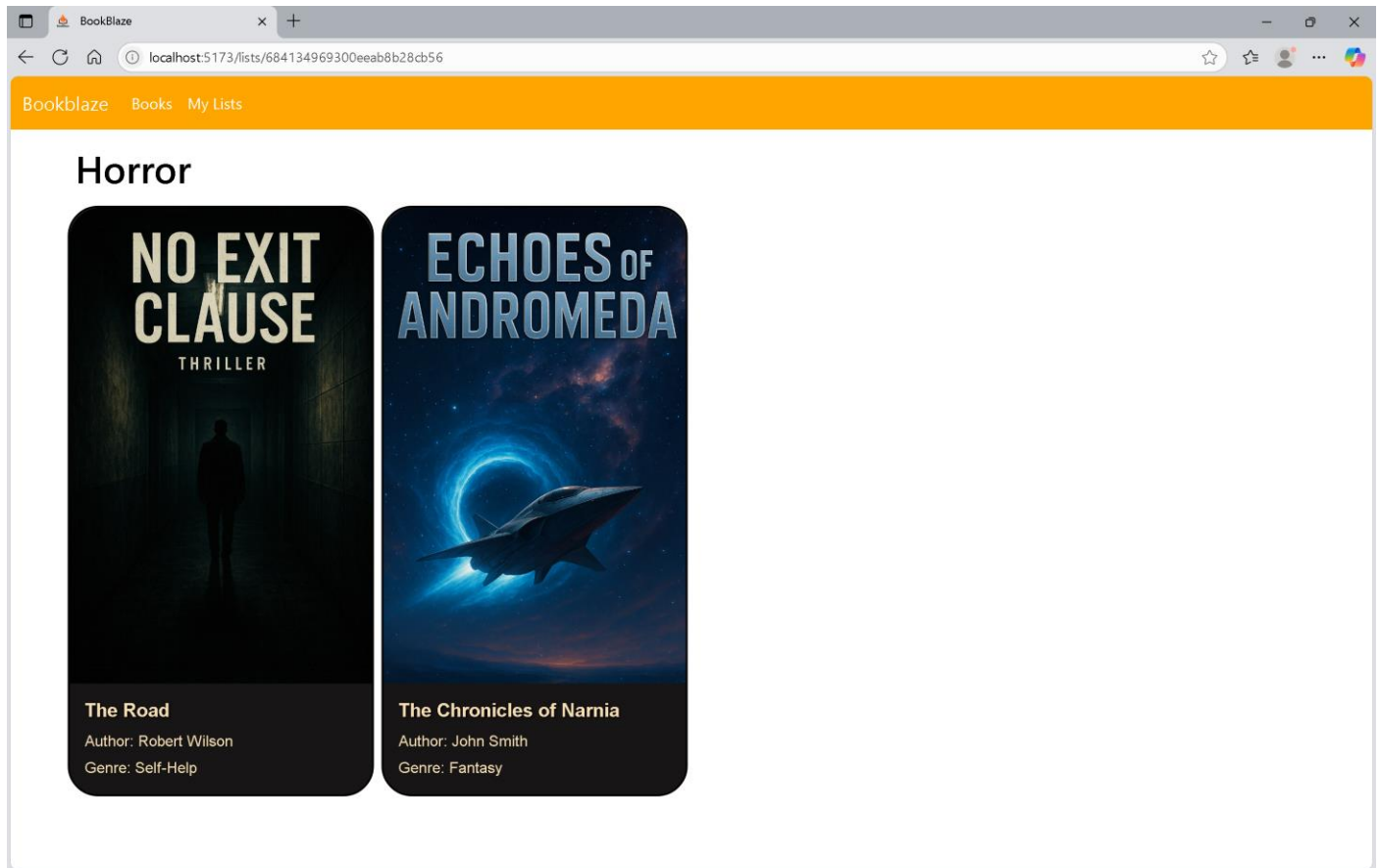
Im Bereich der bestehenden Listen erkennt man für jede Liste eine Karte, auf der der Name der Liste als Überschrift zu sehen ist. Unter dem Listentitel gibt es zwei Möglichkeiten: Ein „View“-Button, um die Inhalte dieser Liste anzusehen, sowie eine „Delete“-Schaltfläche. Klickt man „Delete“, verschwindet die Karte sofort und die Liste wird aus der Anzeige entfernt. Sobald eine neue Liste erstellt oder gelöscht wurde, aktualisiert sich die Anzeige automatisch, sodass immer der aktuelle Stand aller Listen sichtbar ist.

Dateien:

- routes/lists/+page.svelte
- routes/lists/+page.server.js

### 3.7. Detailansicht einer Liste (Single List View)

Route: /lists/[list\_id]



Oben in großer Schrift steht der Name der ausgewählten Leseliste. Darunter folgen alle Bücher, die zu dieser Liste gehören, in einer übersichtlichen Kachelansicht. Jede Kachel zeigt Titel, Autor, Genre und – falls verfügbar – ein Coverbild. Die Kacheln sind in einem flexiblen Raster angeordnet und passen sich automatisch an unterschiedliche Fensterbreiten an. Der Nutzer kann auf eine Buch-Kachel klicken, um direkt zur Detailansicht dieses Buchs zu gelangen (siehe Seite 3.4).

Dateien:

- lib/components/BookCard.svelte
- routes/lists/[list\_id]/+page.svelte
- routes/lists/[list\_id]/+page.server.js



## 4. Erweiterungen

### *Erweiterungen: Edit-Funktion und Listen-Funktion*

#### 4.1. Edit-Funktion

Auf der Buch-Detailseite gibt es einen „Edit Book“-Link, der zu einem Formular führt, in dem Titel, Autor und Genre bereits mit den aktuellen Werten vorbelegt sind. Der Nutzer kann Änderungen vornehmen und mit „Save Changes“ speichern, woraufhin die aktualisierten Daten sofort in der gesamten App übernommen werden. Mit „Cancel“ kehrt man ohne Speicherung zur Detailansicht zurück, sodass sich nichts ändert. Datei(en):

- lib/db.js
- lib/components/BookCard.svelte
- routes/books/[book\_id]/edit/+page.svelte
- routes/books/[book\_id]/edit /+page.server.js

#### 4.2. Listen-Funktion

Nutzer können auf einer Seite einen Namen für eine neue Liste eingeben und aus allen vorhandenen Büchern per Mehrfachauswahl diejenigen auswählen, die in die Liste gehören. Nach Klick auf „Create List“ erscheint die neue Liste sofort in der Übersicht. Jede Liste wird als Kachel mit ihrem Namen angezeigt—dort gibt es einen „View“-Button, um alle darin enthaltenen Bücher als Kacheln anzuzeigen, und einen „Delete“-Button, der die Liste sofort entfernt. In der Detailansicht einer Liste sieht man oben den Listennamen und darunter alle beinhalteten Bücher als BookCards; Klick auf eine BookCard öffnet die jeweilige Buch-Detailseite. Zurück zur Listenübersicht geht man über das Hauptmenü („Lists“) oder den Browser-Zurück-Button.

Dateien:

- lib/db.js
- lib/components/BookCard.svelte
- routes/lists/list\_id]/+page.svelte
- routes/lists/list\_id]/+page.server.js