

Prototyping Projektdokumentation

Name: Denise Aeschlimann

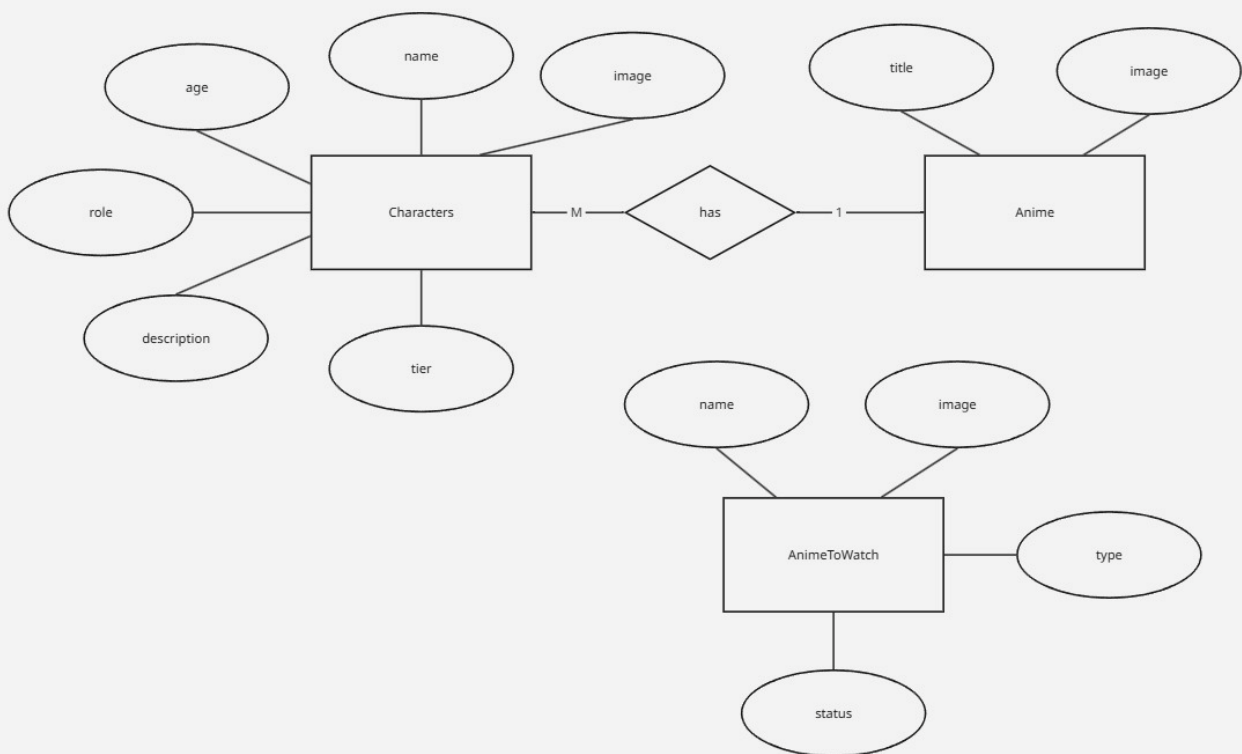
E-Mail: aeschden@students.zhaw.ch

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

1. Einleitung

MyAnimeList basiert auf meinen drei Lieblingsanime und ist eine Webanwendung, mit der Anime-Fans ihre Serien und Filme übersichtlich verwalten. Für jeden der drei Favoriten erscheint eine geordnete Übersicht aller bereits erfassten Charaktere, denen man einen Rang von S bis F zuweist. Über ein Formular lassen sich zusätzliche Informationen wie Name, Alter, Rolle, Beschreibung und ein Bildlink ergänzen. Einzelne Charaktereinträge können im Detail angezeigt und bei Bedarf wieder entfernt werden, um die Liste aktuell zu halten. Ausserdem gibt es eine integrierte Watchlist, in der Serien oder Filme anhand von Titel, Bild, Typ (Serie oder Film) und Status (laufend, beendet, angekündigt) gespeichert und direkt auf der Seite abgerufen werden. Im Hintergrund sorgt MongoDB für die dauerhafte Speicherung aller Daten, während SvelteKit 5 mit der neuen Runen-API reaktive Datenflüsse und eine schnelle Benutzeroberfläche ermöglicht.

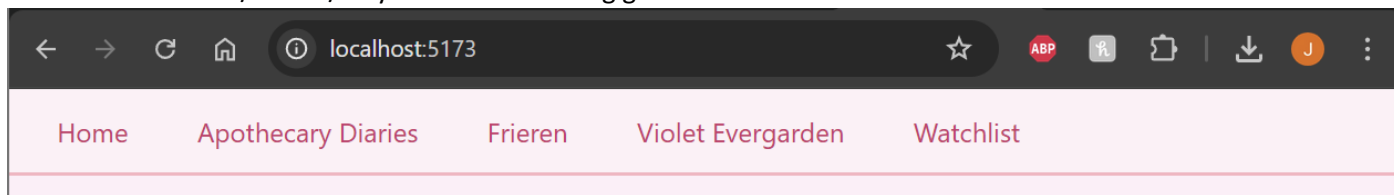
2. Datenmodell



3. Beschreibung der Anwendung

3.1. Navigation

Position: wird in `src/routes/+layout.svelte` einmalig geladen und umschließt alle Seiten.

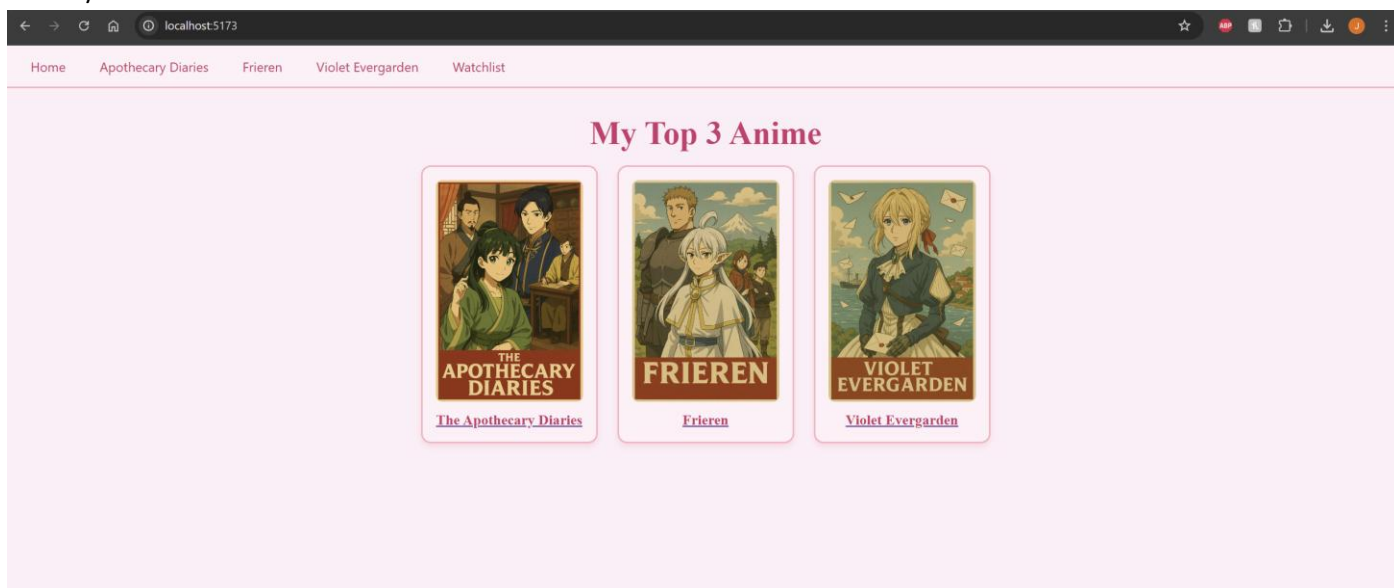


Die Komponente ermöglicht nahtlosen Wechsel zwischen Startseite, den drei Anime-Tierlisten und der Watchlist.
Datei(en):

- `src/lib/components/NavBar.svelte`
- `src/routes/+layout.svelte`

3.2. Startseite (Anime-Übersicht)

Route: /



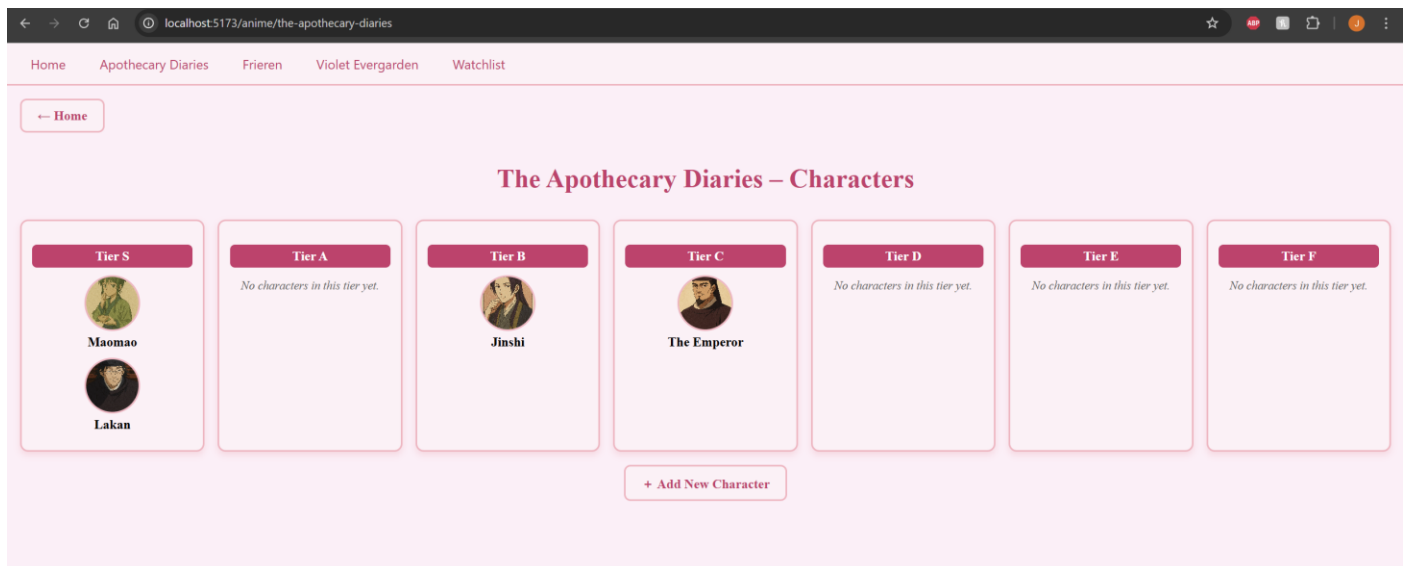
Auf dieser Seite werden meine drei Favoriten als klickbare Karten angezeigt. Durch einen Klick auf eine Karte gelangt man zur entsprechenden Tier-Listen-Ansicht des gewählten Anime.

Datei(en):

- `src/routes/+page.server.js`
- `src/routes/+page.svelte`

3.3. Tier-Listen-Ansicht

Route: /anime/[slug]



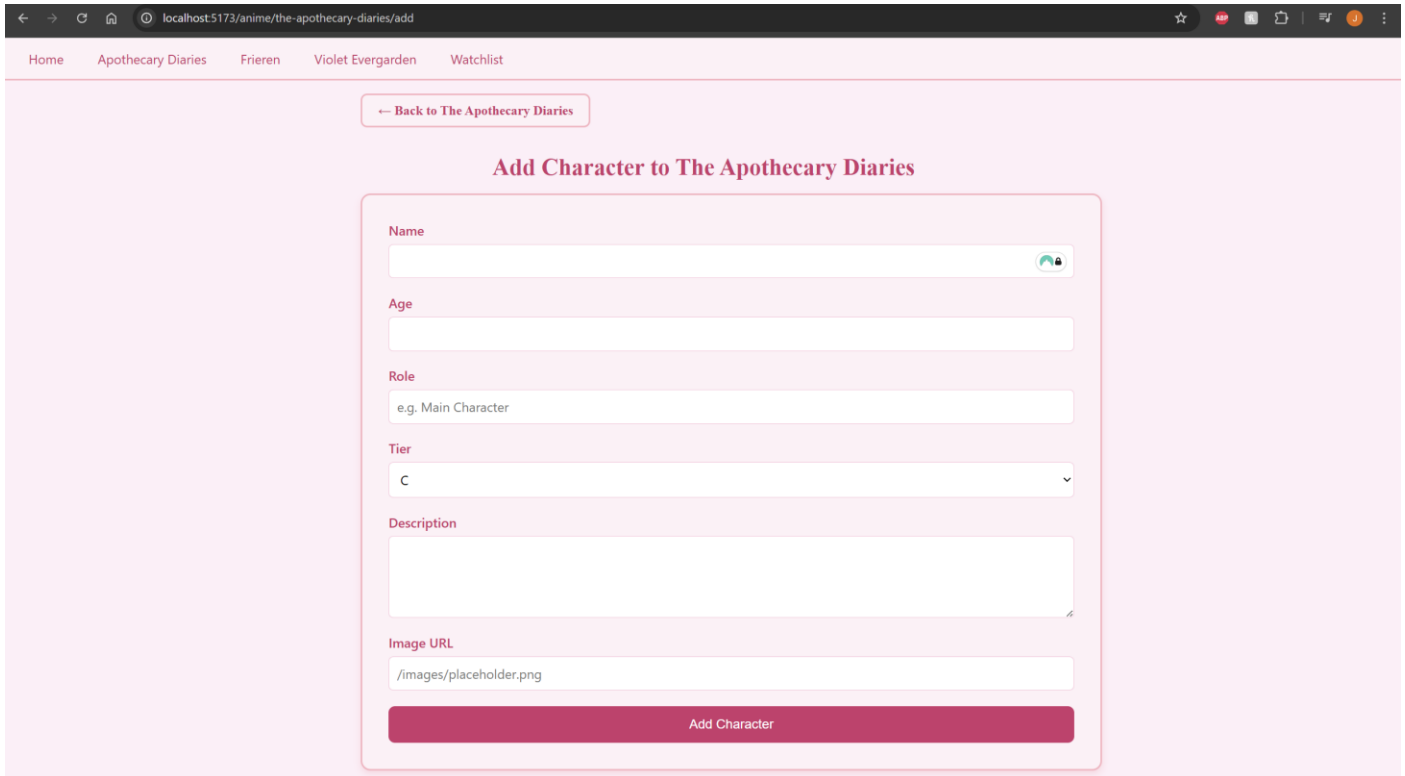
Auf dieser Seite werden alle bereits erfassten Charaktere eines Anime übersichtlich nach den Rängen S bis F gruppiert dargestellt. Die Seite nutzt den URL-Parameter slug (z. B. the-apothecary-diaries), um das jeweilige Anime-Dokument aus der Datenbank zu laden. Für jede Tier-Kategorie zeichnet die Komponente TierList.svelte eine Spalte, in der die einzelnen Charakter-Karten erscheinen. Über einen Klick auf einen Charakter in der Liste gelangt man zur Detailansicht, und mit dem Button «+ Add New Character» wechselt man zur Formularseite für genau diesen slug.

Datei(en):

- src/routes/anime/[slug]/+page.server.js
- src/routes/anime/[slug]/+page.svelte
- src/lib/components/TierList.svelte

3.4. Charakter hinzufügen

Route: `/anime/[slug]/add`



The screenshot shows a web browser at the URL `localhost:5173/anime/the-apothecary-diaries/add`. The page has a pink header with navigation links: Home, Apothecary Diaries, Frieren, Violet Evergarden, and Watchlist. Below the header is a button labeled '← Back to The Apothecary Diaries'. The main content area is titled 'Add Character to The Apothecary Diaries' and contains a form with the following fields:

- Name**: A text input field with a user icon on the right.
- Age**: A text input field.
- Role**: A text input field with the placeholder text 'e.g. Main Character'.
- Tier**: A dropdown menu currently showing 'C'.
- Description**: A large text area for a detailed description.
- Image URL**: A text input field with the placeholder text '/images/placeholder.png'.

At the bottom of the form is a red button labeled 'Add Character'.

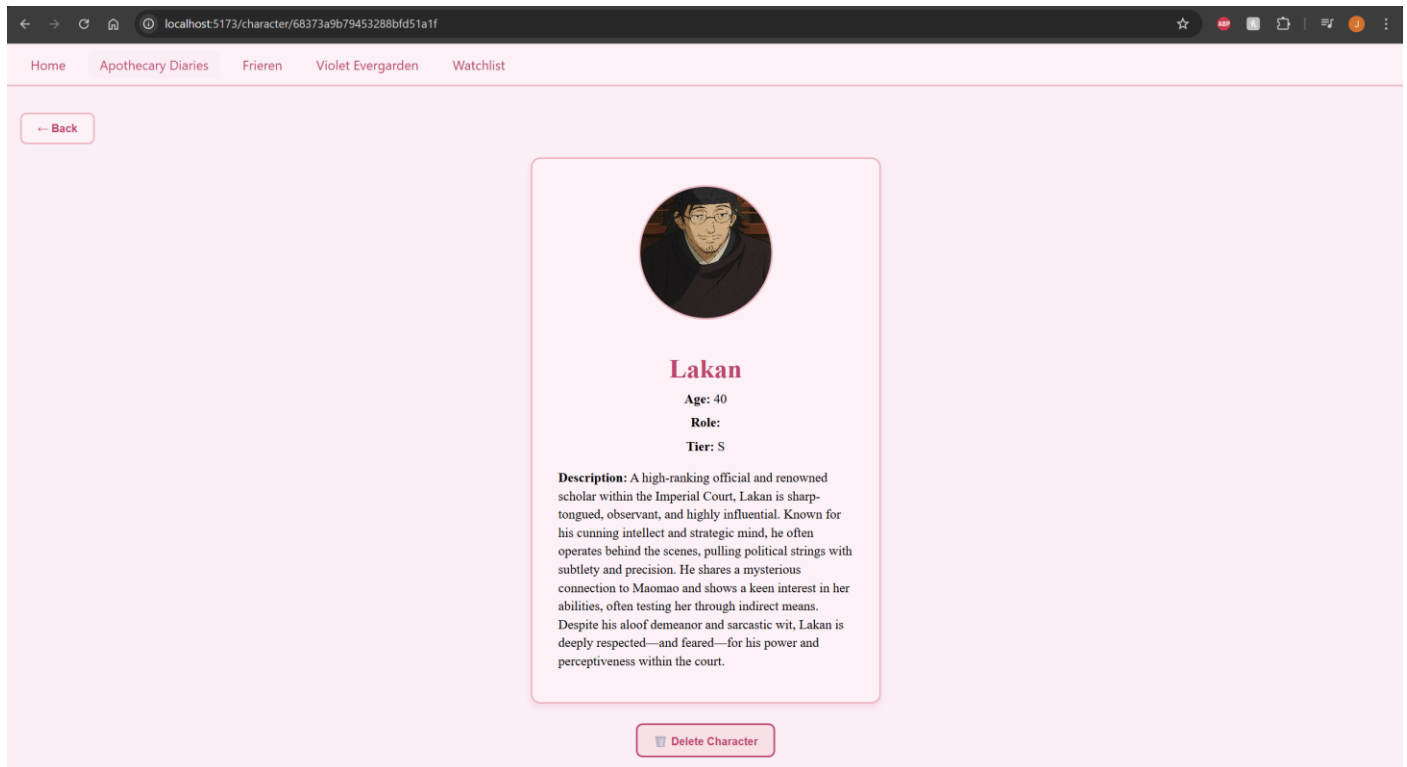
Diese Seite stellt ein Formular bereit, um einem ausgewählten Anime neue Charaktere hinzuzufügen. Sie wird ebenfalls über den slug im Pfad adressiert, sodass beim Absenden automatisch ein Eintrag mit diesem animeSlug in der Datenbank landet. Die Komponente `CharacterForm.svelte` übernimmt alle Eingabefelder (Name, Alter, Rolle, Tier, Beschreibung, Bild-URL). Nach erfolgreicher Validierung und Speicherung erfolgt eine Weiterleitung zurück zur Tier-Listen-Ansicht des betreffenden slug.

Datei(en):

- `src/routes/anime/[slug]/add/+page.server.js`
- `src/routes/anime/[slug]/add/+page.svelte`
- `src/lib/components/CharacterForm.svelte`

3.5. Charakter-Detailseite

Route: /character/[id]



Detailsansicht eines einzelnen Charakters mit allen gespeicherten Feldern und Bild. Bietet „Back“-Button zur Rückkehr und „Delete“-Button zum Entfernen des Charakters aus der Datenbank. Nach Löschen erfolgt Redirect zurück zur Tierlistenseite.

Datei(en):

- src/routes/character/[id]/+page.server.js
- src/routes/character/[id]/+page.svelte

3.6. Watchlist

Route: /watchlist

The screenshot shows a web browser window with the address bar displaying 'myanimelistdenise.netlify.app/watchlist'. The page has a light pink background and is titled 'My Watchlist' in a dark pink font. Below the title, there are two filter sections: 'Filter by Type:' with checkboxes for 'Series' and 'Movie' (both checked), and 'Filter by Status:' with checkboxes for 'Ongoing', 'Done', and 'Announced' (all checked). The main content area displays three anime entries, each in a rounded rectangular card. The first card is for 'One Piece' with a 'ONE PIECE' logo, 'Type: series', and 'Status: ongoing'. The second card is for 'dawdawd' with an 'ANIME IMAGE' placeholder, 'Type: series', and 'Status: ongoing'. The third card is for 'test' with an 'ANIME IMAGE' placeholder, 'Type: movie', and 'Status: announced'. Below the list, there is an 'Add to Watchlist' section containing a form with fields for 'Title', 'Image URL' (with a placeholder '/images/placeholder.png'), 'Type' (a dropdown menu set to '-- Select type --'), and 'Status' (a dropdown menu set to '-- Select status --'). An 'Add Anime' button is located at the bottom right of the form.

Übersichts- und Eingabeseite für beliebige Anime oder Filme. Jeder Eintrag enthält Titel, Bild, Typ (Serie/Film) und Status (laufend/beendet/angekündigt). Neue Einträge lassen sich über ein Formular hinzufügen und erscheinen sofort in der Liste.

Datei(en):

- src/lib/components/EntrCard.svelte
- src/routes/watchlist/+page.svelte
- src/routes/watchlist/+page.server.js

4. Erweiterungen

4.1. Integrierte Watchlist als zweite Entität

Auf der Seite /movies können via Checkbox Filme ausgeblendet werden, die nicht auf der Watchlist stehen. Dies wurde mit einem Formular und einer MongoDB-Query in der Funktion `getMoviesOnWatchlist()` in der Datei `lib/db.js` realisiert.

Datei(en):

- `src/lib/components/EntryCard.svelte`
- `src/routes/watchlist/+page.svelte`
- `src/routes/watchlist/+page.server.js`

4.2. Filterfunktion auf der Watchlist-Seite

Auf der Route /watchlist gibt es Checkboxes, mit denen Nutzer Einträge nach Typ (series oder movie) und Status (ongoing, done, announced) auswählen können. Die Filterung passiert komplett clientseitig: Hinter den Checkboxes stecken Runes-States, aus denen eine abgeleitete Liste generiert wird. Sobald eine Auswahl geändert wird, aktualisiert sich diese Liste automatisch und zeigt nur noch die passenden Einträge an.

Datei(en):

- `src/lib/components/EntrCard.svelte`
- `src/routes/watchlist/+page.svelte`
- `src/routes/watchlist/+page.server.js`

4.3. Hover-Effekte auf Character-Karten

Beim Überfahren („hover“) hebt sich jede Tier-Sektion mit `transform: translateY()` und stärkerem `box-shadow` hervor.

Datei(en):

- `src/lib/components/TierList.svelte`

4.4. Automatische Tier-Positionierung

Charaktere landen beim Laden oder Hinzufügen ohne manuelles Sortieren automatisch in der richtigen Kategorie (S–F). Die Komponente `TierList.svelte` gruppiert alle Einträge reaktiv anhand ihres `tier`-Feldes und zeigt sie je in der passenden Sektion an.

Datei(en):

- `src/lib/components/TierList.svelte`

4.5. Charakter löschen

Auf der Detailseite kann ein Charakter per POST-Formular gelöscht und danach automatisch zurück zur Tierliste umgeleitet werden.

Datei(en):

- `src/routes/character/[id]/+page.server.js`
- `src/routes/character/[id]/+page.svelte`