

Prototyping Projektdokumentation

Name: **Lukas Oetiker**

E-Mail: **oetiklu1@students.zhaw.ch**

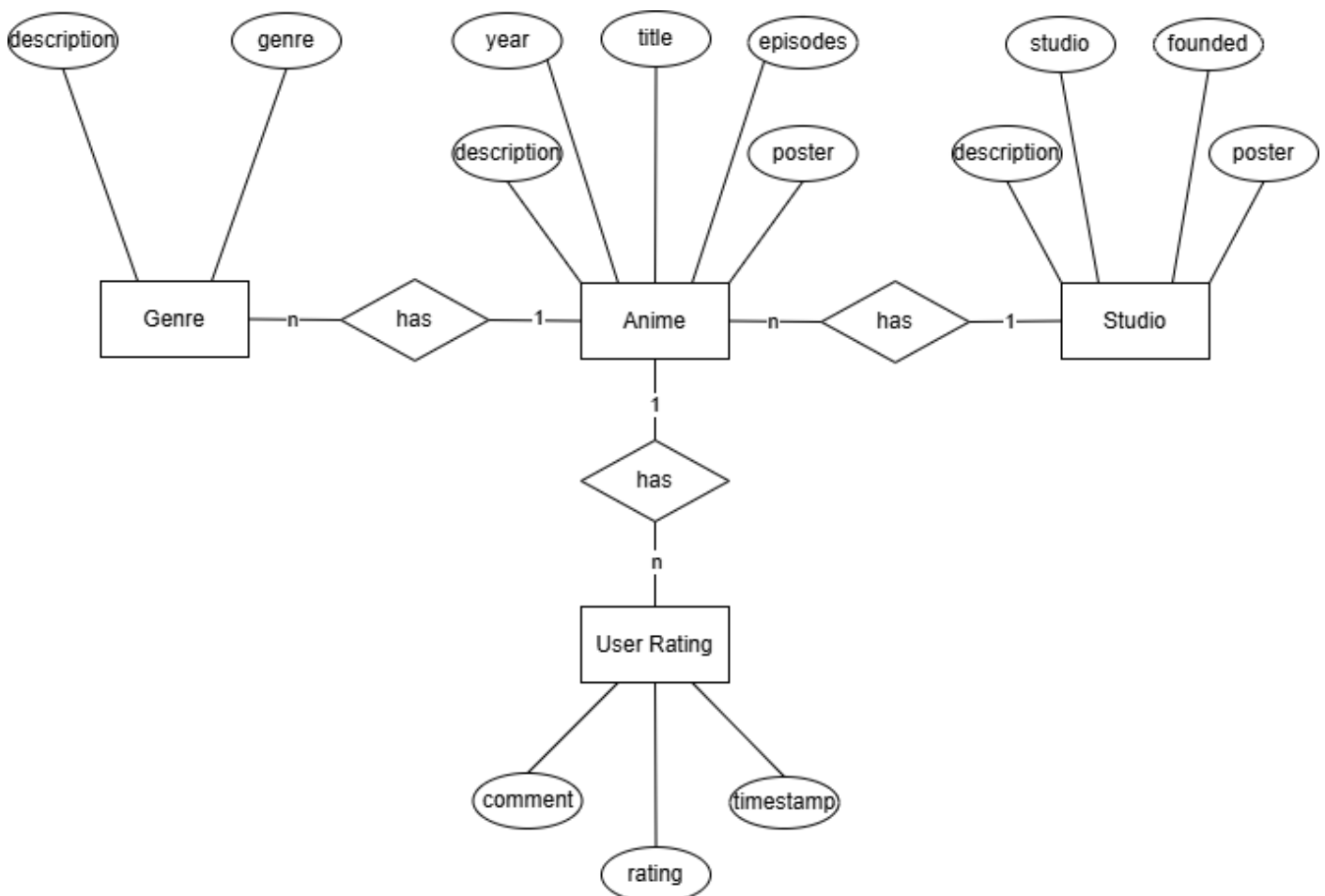
URL der deployten Anwendung: **<https://anibuddy-1.netlify.app/>**

Einleitung

Von einer nerdigen Randerscheinung hin zu einer der beliebtesten Formen, Filme und Serien des 21. Jahrhunderts zu konsumieren – Anime. Der Begriff Anime hat in der westlichen Welt zunehmend an Bedeutung gewonnen. Früher war es notwendig, auf spezifische Fernsehsender oder fragwürdige Websites zurückzugreifen, um Anime zu schauen. Heute hingegen sind Anime-Serien und -Filme auf den populärsten Streaming-Plattformen wie Netflix, Disney+ und Amazon Prime verfügbar. Darüber hinaus gibt es spezialisierte Dienste wie Crunchyroll, die durch diesen Boom erheblich profitieren konnten.

Anibuddy ist da, um neugierigen Menschen den Einstieg in die vielseitige Welt der Anime zu erleichtern. Die Anwendung bietet Nutzern einen Überblick über beliebte Animes, liefert detaillierte Beschreibungen und ermöglicht es, mehr über die unterschiedlichen Studios und Branchen zu erfahren. Welche Studios prägen die Anime-Welt? Zudem können Nutzer Bewertungen abgeben, um anderen Anime-Neulingen authentische Rezensionen bereitzustellen. So wird eine Alternative zu den oft voreingenommenen Bewertungen von eingefleischten Anime-Fans geschaffen.

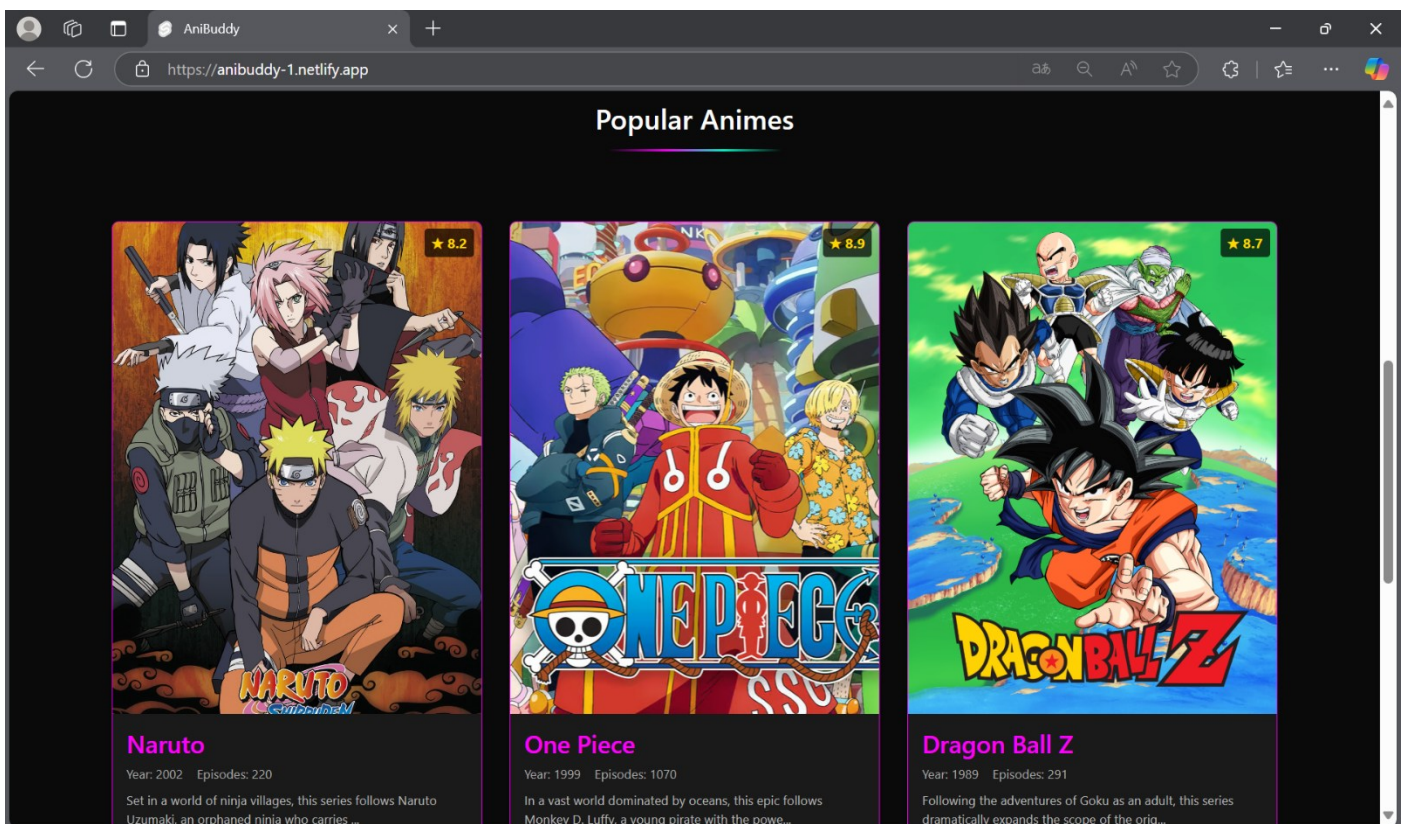
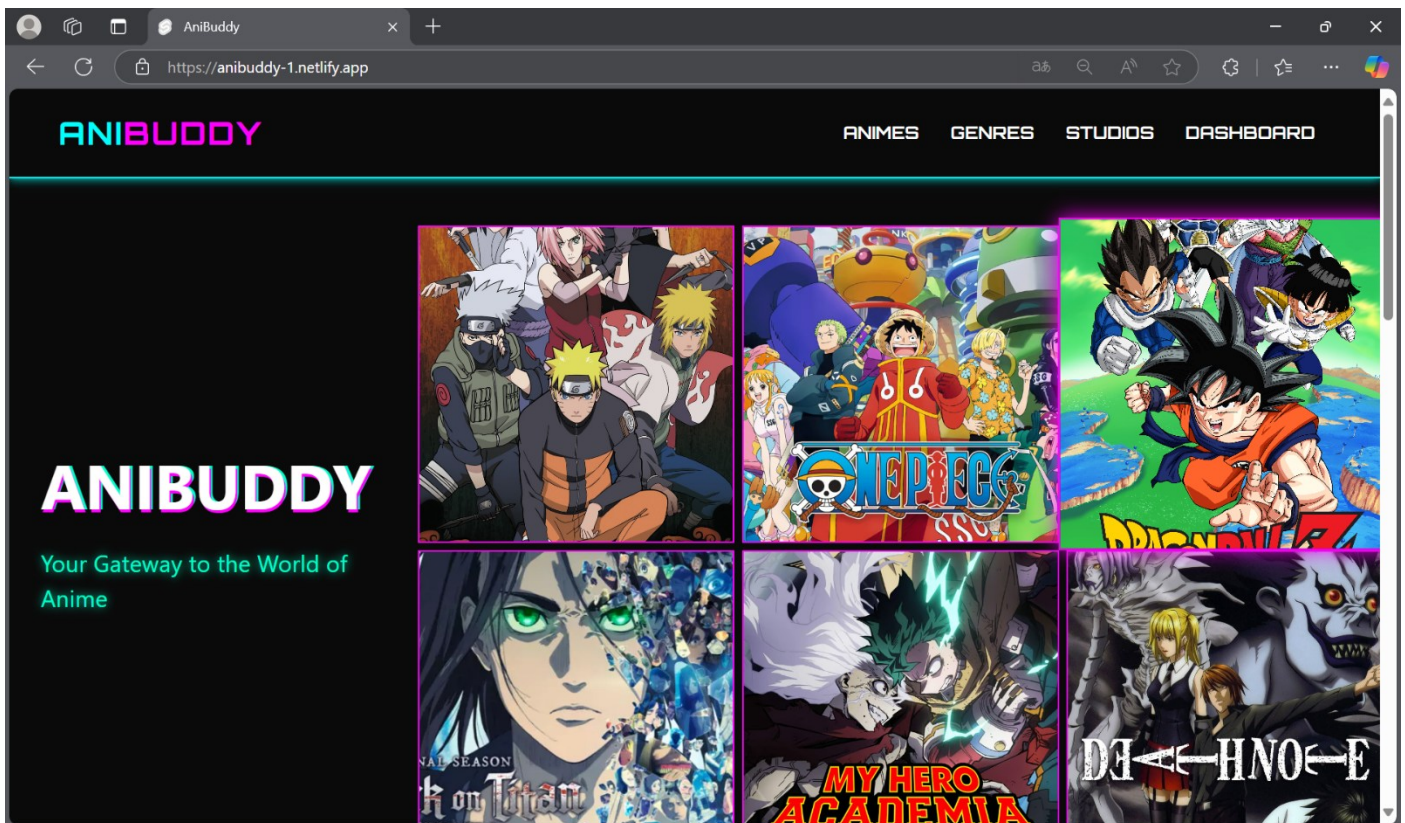
Datenmodell

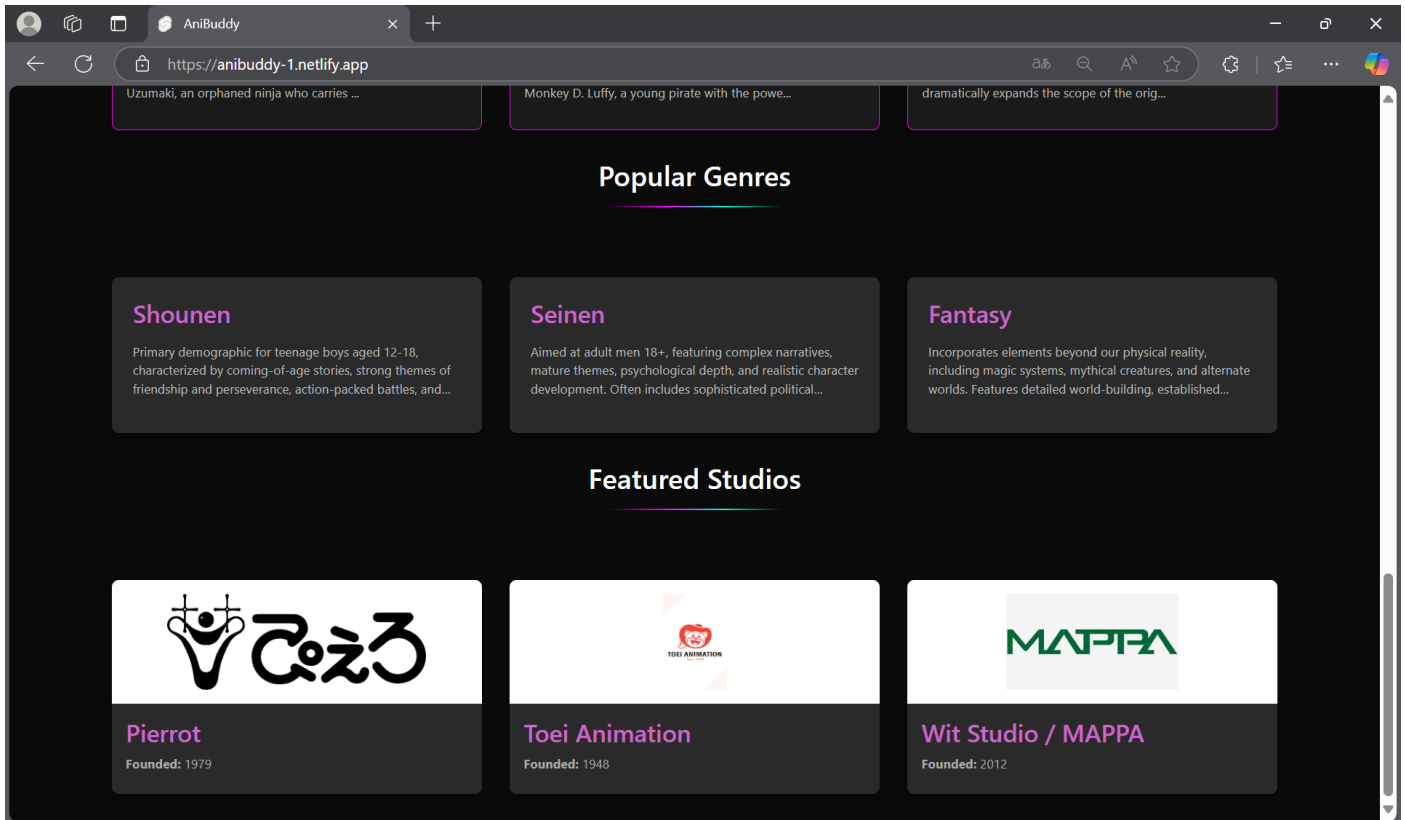


Beschreibung der Anwendung

Hauptseite

Route: /src/routes





Auf dieser Seite wird der Benutzer mit Bildern von Animes begrüßt, die aus dem Verzeichnis images stammen. Der auffällige «Anibuddy»-Schriftzug wird durch die CSS-Klasse «glitch» erzeugt. Weiter unten auf der Seite werden die «Popular Animes» angezeigt, die durch die Komponente AnimeCard1.svelte dargestellt werden. Die Animes sind auswählbar und führen zu /animes/[anime_id].

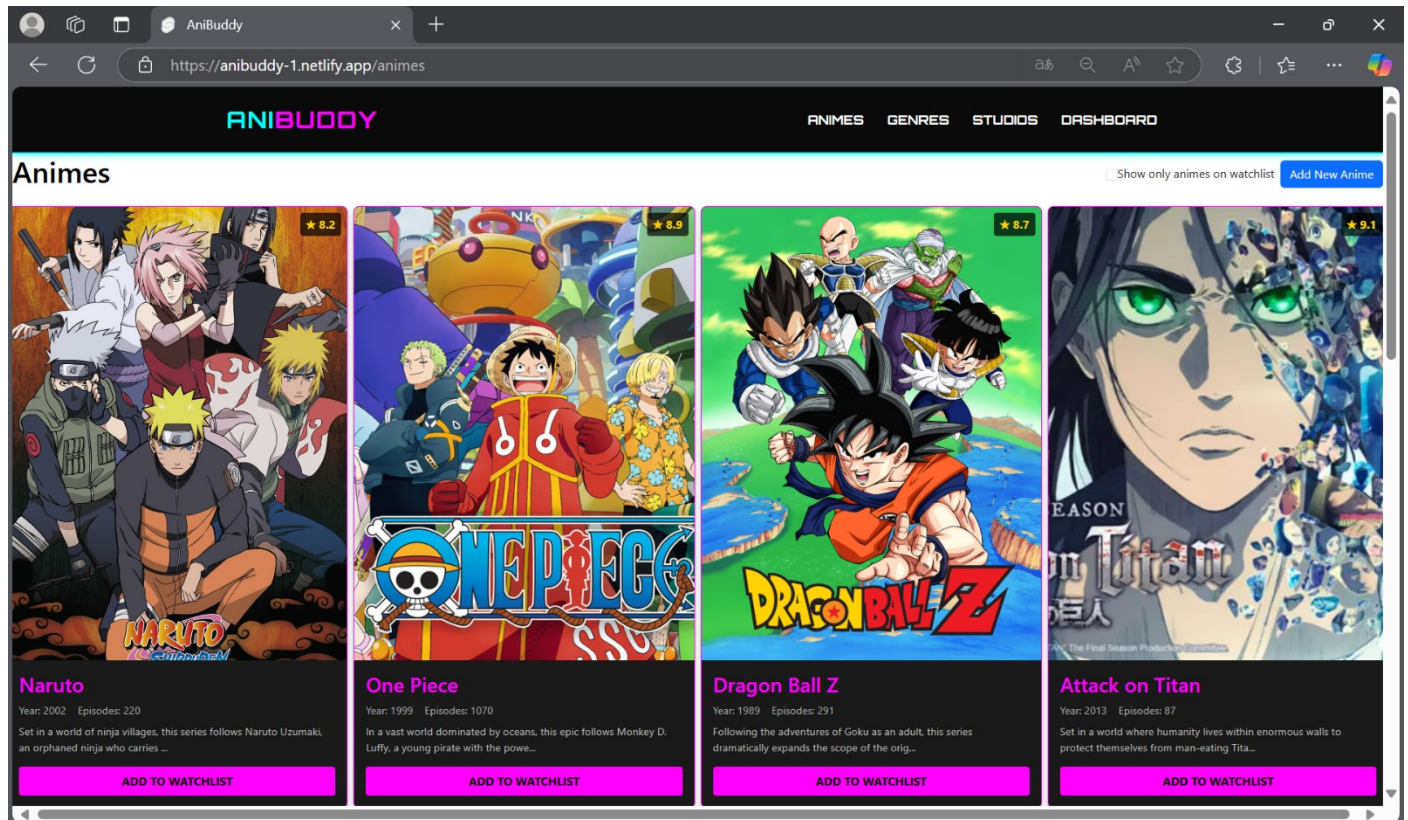
Das gleiche Prinzip gilt für Genres und Studios, die jeweils durch die Komponenten GenreCard.svelte und StudioCard.svelte dargestellt werden. Über die Navigation oben rechts können die Seiten /animes, /genres und /studios erreicht werden.

Dateien:

- lib/components/AnimeCard1.svelte
- lib/components/GenreCard.svelte
- lib/components/StudioCard.svelte
- routes/page.svelte
- routes/page.server.js

Animes

Route: /src/routes/animes



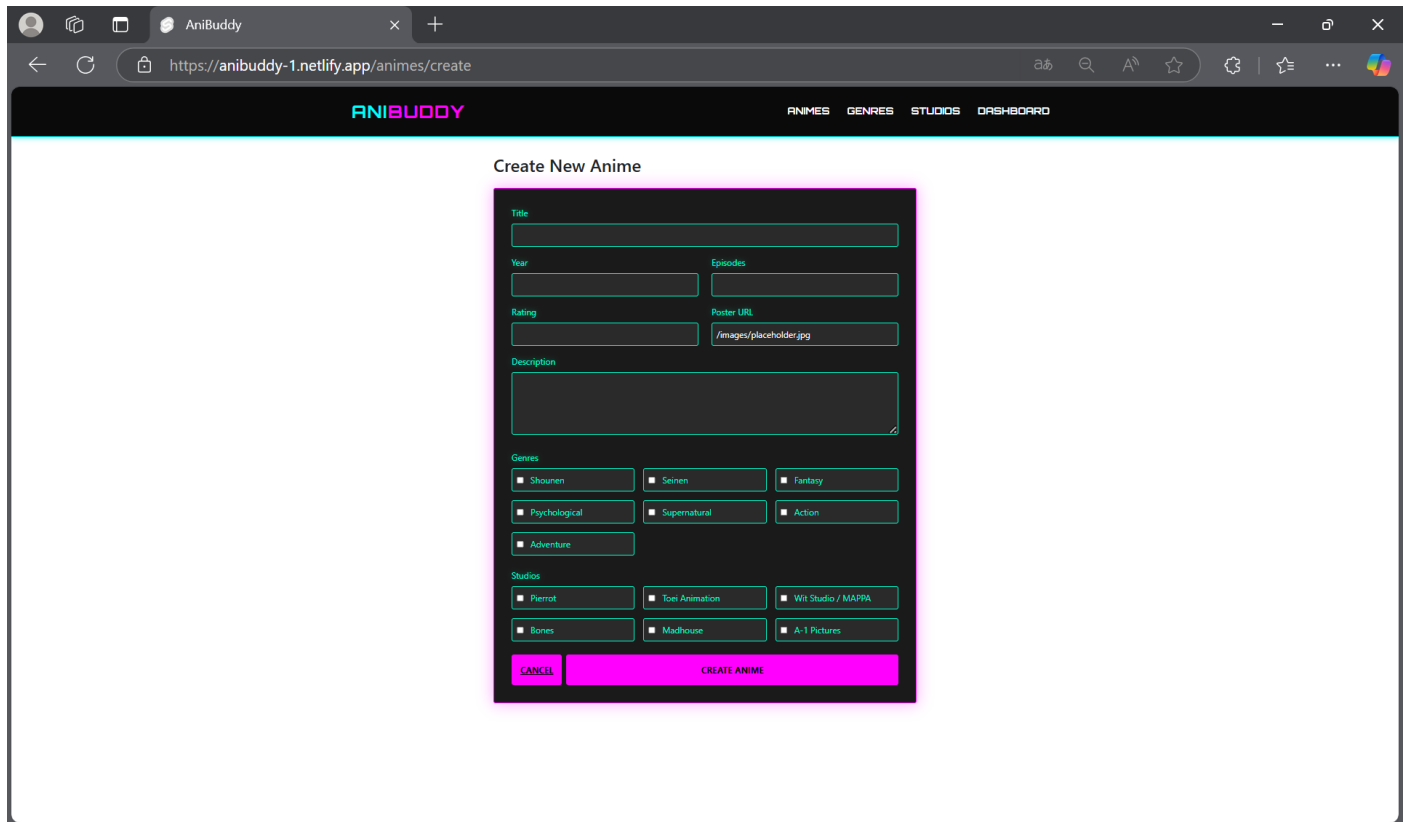
Auf dieser Seite werden alle Animes aufgelistet. Die Darstellung der einzelnen Animes erfolgt durch die Komponente `AnimeCard.svelte`, die sich von `AnimeCard1.svelte` durch die Funktion „Add to Watchlist“ unterscheidet. Animes können zur Watchlist hinzugefügt oder daraus entfernt werden. Mit der Checkbox «Show only animes on watchlist» können alle Animes, die nicht auf der Watchlist sind, ausgeblendet werden. Über den Button «Add New Anime» gelangt man zur Seite `/animes/create`. Ein Klick auf einen Anime führt zur Detailseite des entsprechenden Anime.

Dateien:

- `lib/components/AnimeCard.svelte`
- `routes/animes/+page.svelte`
- `routes/animes/+page.server.js`

Animes Create

Route: `/src/routes/animes/create`



The screenshot shows a web browser window with the URL `https://anibuddy-1.netlify.app/animes/create`. The page title is "ANIBUDDY" and the navigation bar includes links for "ANIMES", "GENRES", "STUDIOS", and "DASHBOARD". The main content area is titled "Create New Anime" and contains a form with the following fields:

- Title:
- Year:
- Episodes:
- Rating:
- Poster URL:
- Description:
- Genres: ☐ Shounen, ☐ Seinen, ☐ Fantasy, ☐ Psychological, ☐ Supernatural, ☐ Action, ☐ Adventure
- Studios: ☐ Pierrot, ☐ Toei Animation, ☐ Wit Studio / MAPPA, ☐ Bones, ☐ Madhouse, ☐ A-1 Pictures

At the bottom of the form are two buttons: "CANCEL" and "CREATE ANIME".

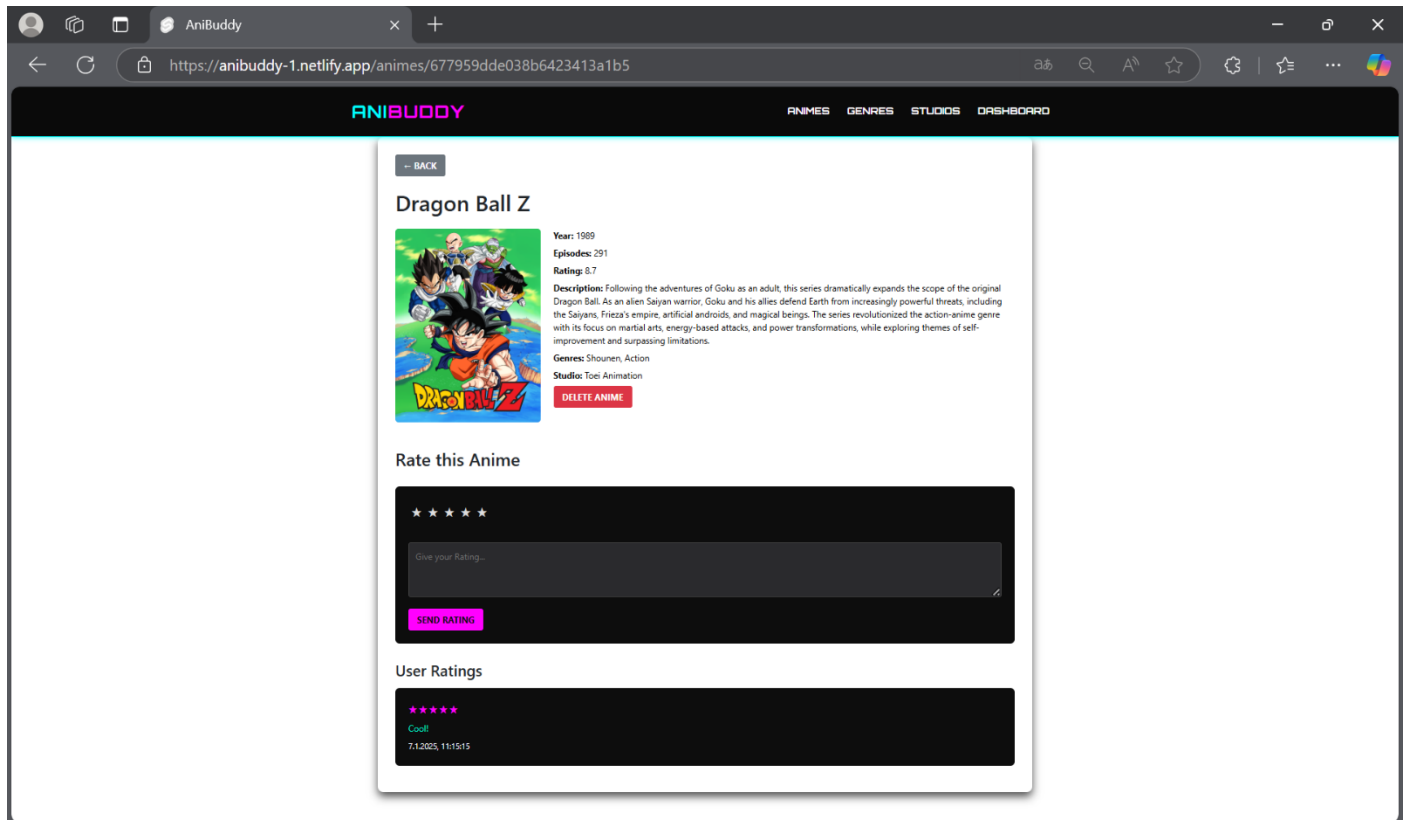
Auf der Seite «Add New Anime» können neue Animes erstellt werden. Die Informationen für den Anime werden in die entsprechenden Felder eingegeben. Mit dem Button «CANCEL» kann das Formular verlassen werden, während der Button «CREATE ANIME» den Anime erstellt. Nach dem Erstellen gelangt man zum neuen Objekt.

Dateien:

- `routes/animes/create/+page.svelte`
- `routes/animes/create/+page.server.js`

Animes Detailseite

Route : `/src/routes/animes/[anime_id]`



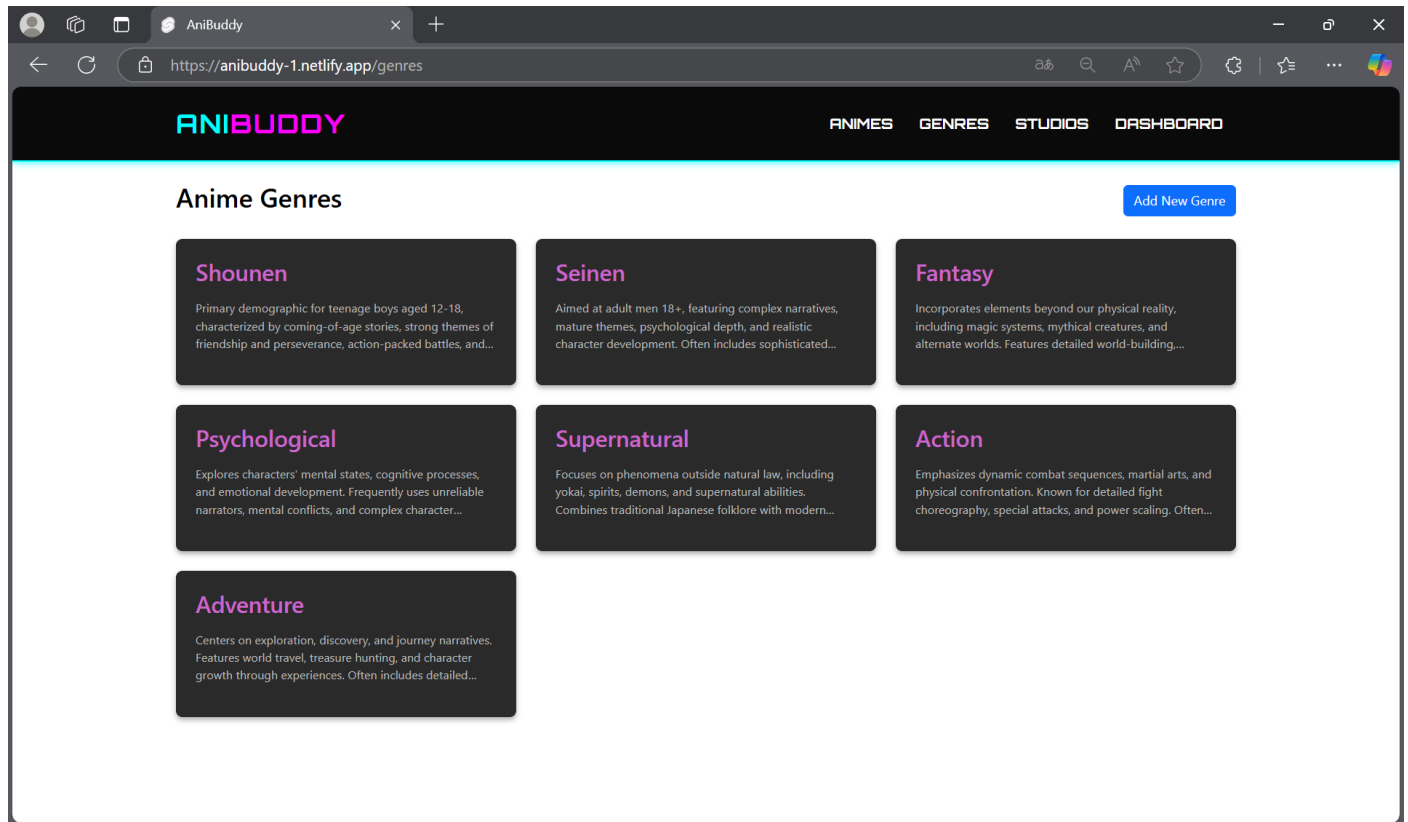
Auf der Detailseite sieht man ein Bild des Anime sowie die dazugehörigen Informationen. Mit dem Button «BACK» gelangt man zurück zur Seite `/animes`. Der Button «DELETE ANIME» löscht den Anime und führt ebenfalls zurück zur Seite `/animes`. Der Anime kann bewertet werden, und das User Rating wird gespeichert und angezeigt, nachdem man den Button «SEND RATING» gedrückt hat.

Dateien:

- `lib/components/RatingSystem.svelte`
- `routes/animes/[anime_id]/+page.svelte`
- `routes/animes/[anime_id]/+page.server.js`

Genres

Route : `/src/routes/genres`



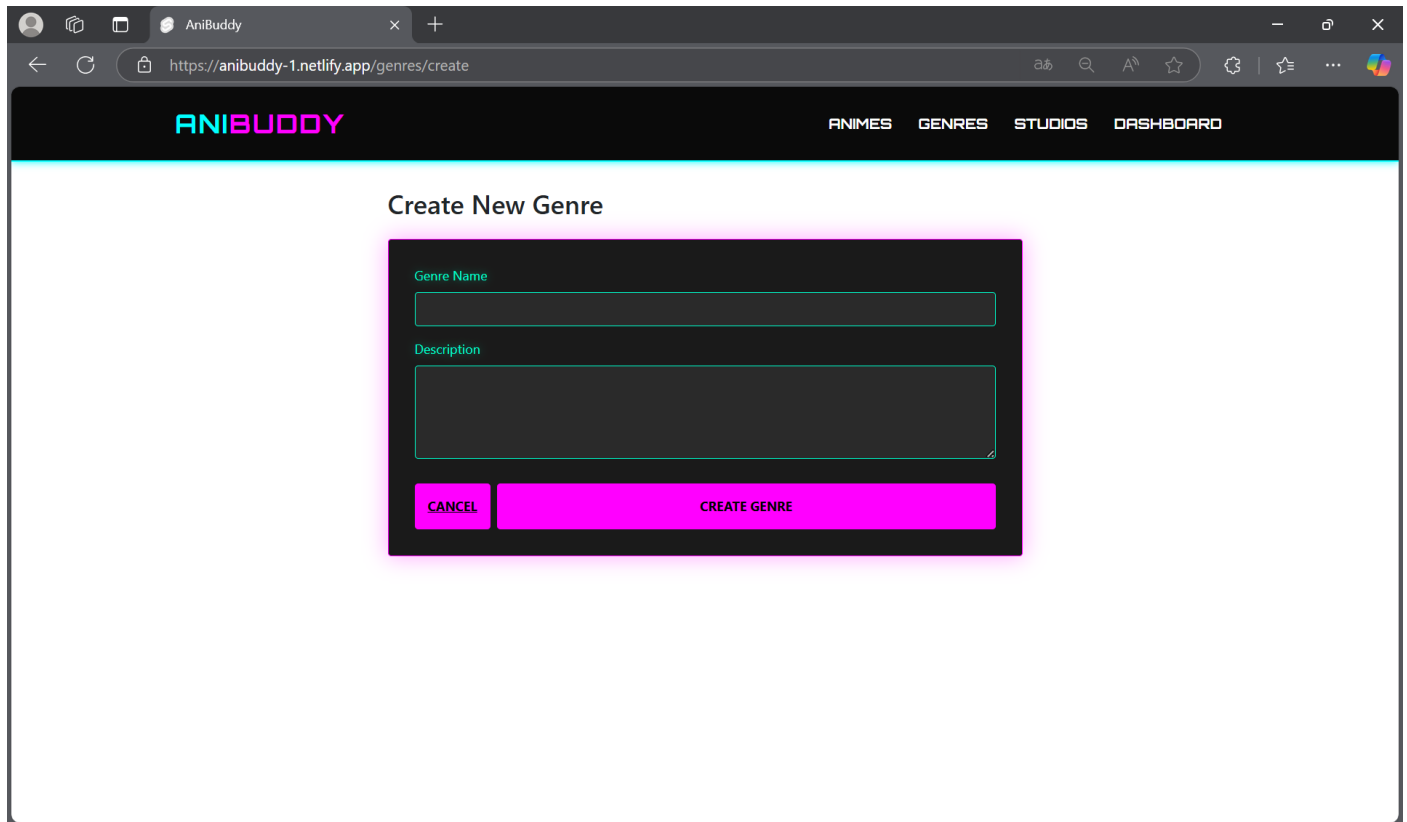
Auf dieser Seite werden alle Genres aufgelistet. Die Darstellung der einzelnen Genres verwendet die `GenreCard.svelte`. Über den Button «Add New Genre» gelangt man zur Seite `/genres/create`. Mit einem Klick auf den Genres landet man auf die Detailseite der entsprechenden Genre.

Dateien:

- `lib/components/GenreCard.svelte`
- `routes/genres/+page.svelte`
- `routes/genres/+page.server.js`

Genres Create

Route: /src/routes/genres/create



The screenshot shows a web browser window with the URL `https://anibuddy-1.netlify.app/genres/create`. The page has a dark theme with a black header bar. The header contains the 'ANIBUDDY' logo in red and blue, and navigation links for 'ANIMES', 'GENRES', 'STUDIOS', and 'DASHBOARD'. The main content area is white and features a 'Create New Genre' form. The form is a white box with a light gray border. It contains two input fields: 'Genre Name' (a single-line text input) and 'Description' (a multi-line text area). Below the inputs are two buttons: a red 'CANCEL' button and a green 'CREATE GENRE' button.

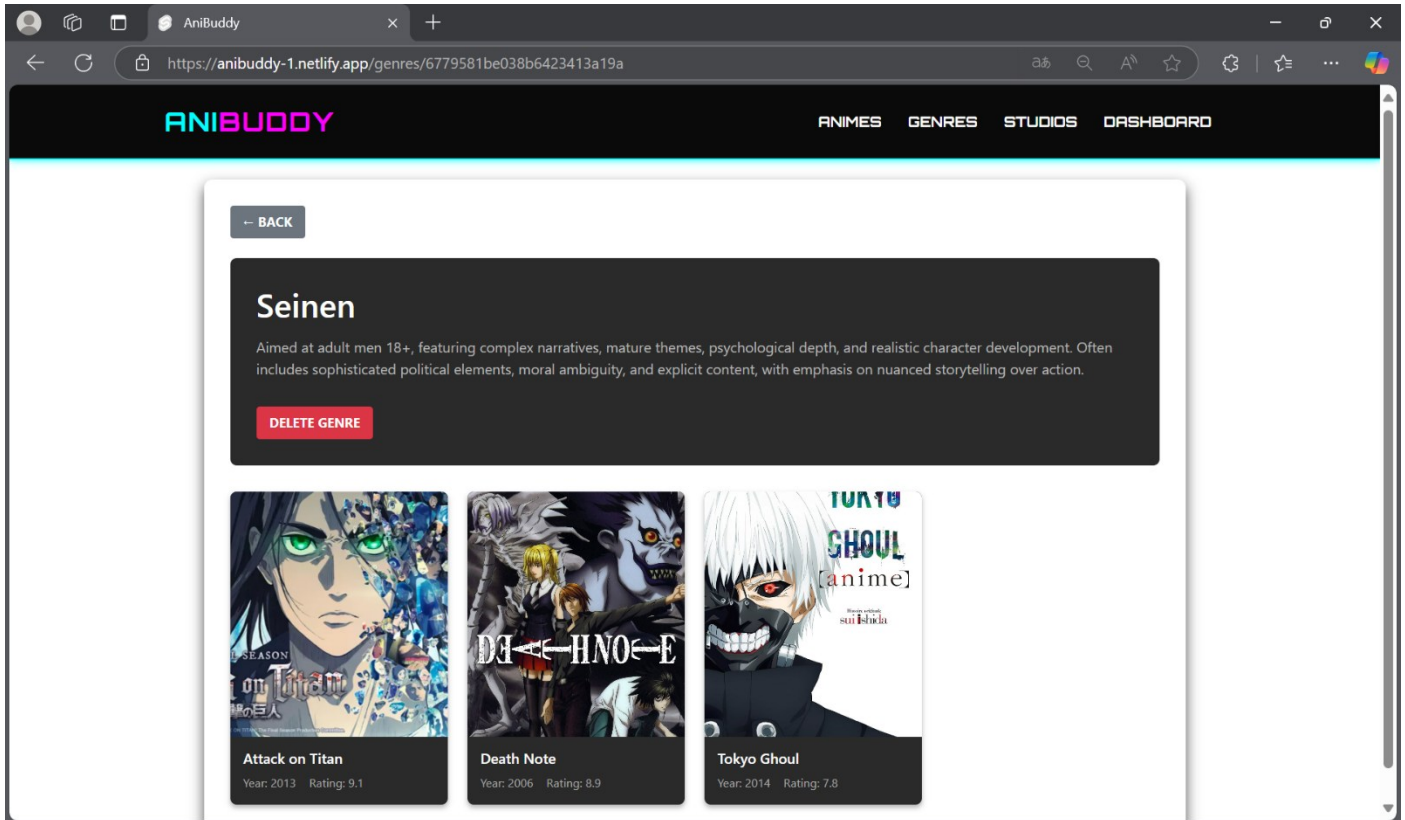
Auf der Seite «Add New Genre» können neue Genres erstellt werden. Die Informationen für den Genre werden in die entsprechenden Felder eingegeben. Mit dem Button «CANCEL» kann das Formular verlassen werden, während der Button «CREATE GENRE» das Genre erstellt. Nach dem Erstellen gelangt man zum neuen Objekt.

Dateien:

- `routes/genres/create/+page.svelte`
- `routes/genres/create/+page.server.js`

Genres Detailseite

Route: `/src/routes/genres/[genre_id]`



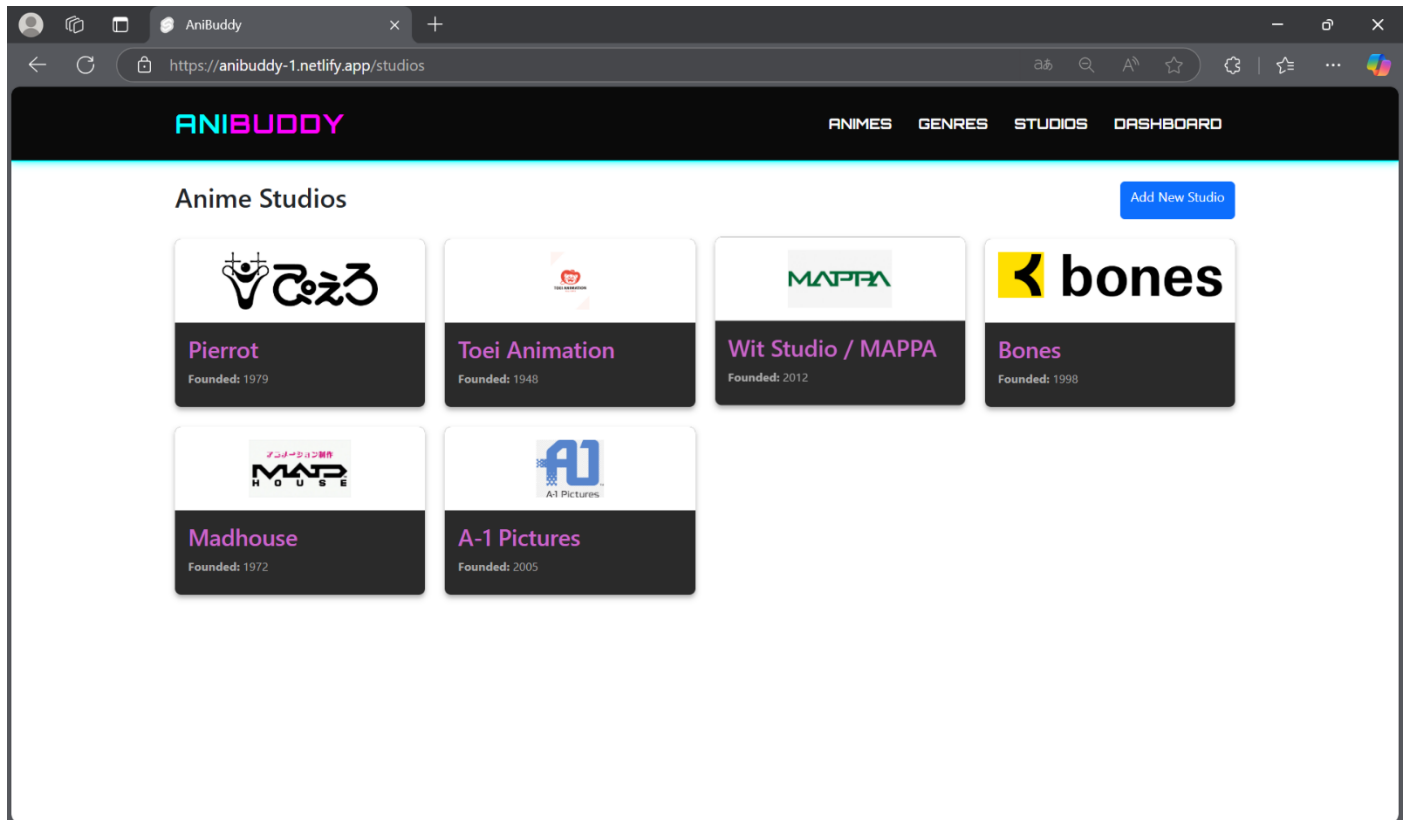
Auf der Detailseite sieht man ein Bild des Genres sowie die dazugehörigen Informationen. Mit dem Button «BACK» gelangt man zurück zur Seite `/genres`. Der Button «DELETE GENRE» löscht das Genre und führt ebenfalls zurück zur Seite `/genres`. Die Animes mit dem entsprechenden Genre werden angezeigt und sind klickbar.

Dateien:

- `routes/animes/[genre_id]/+page.svelte`
- `routes/animes/[genre_id]/+page.server.js`

Studios

Route: /src/routes/studios



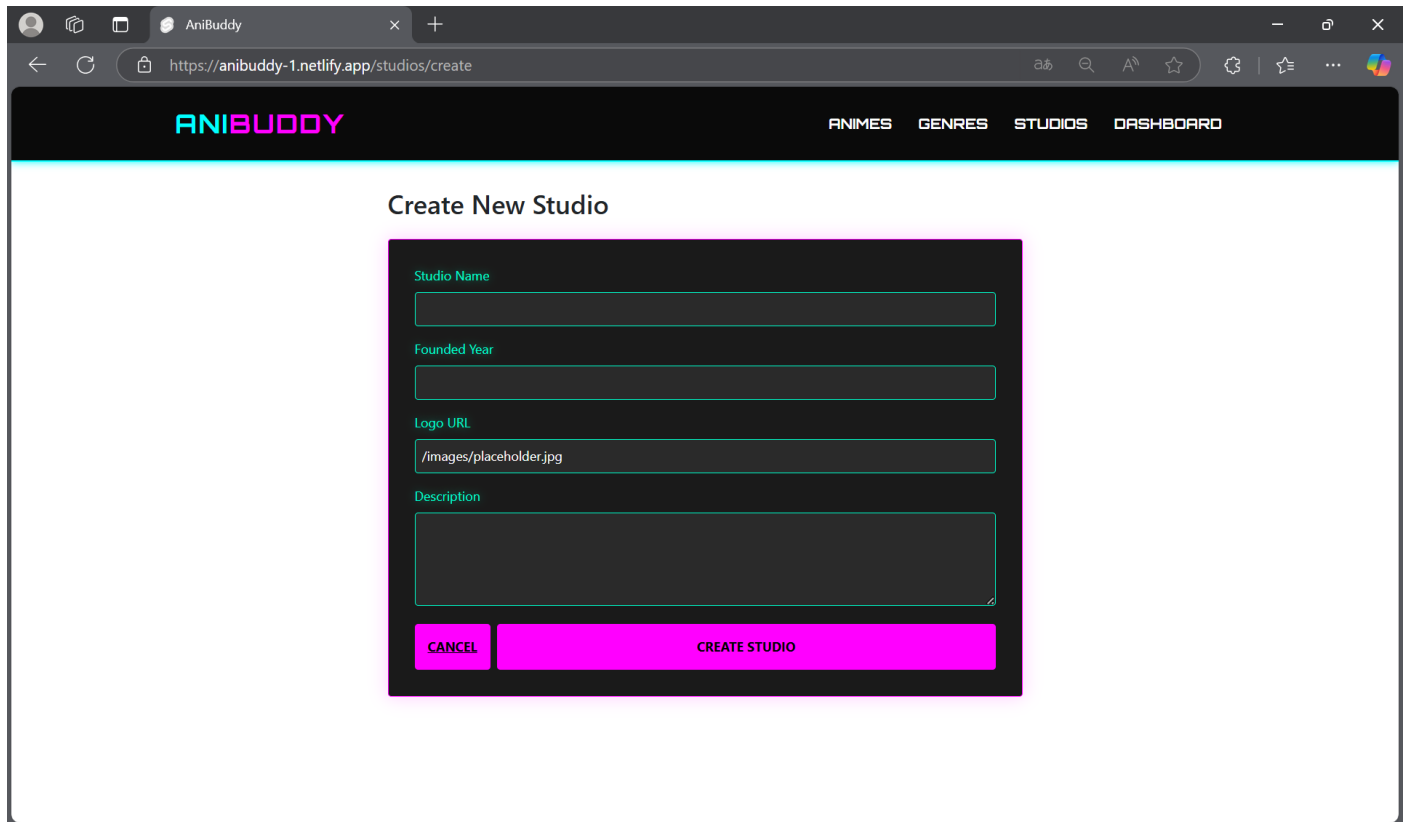
Auf dieser Seite werden alle Studios aufgelistet. Die Darstellung der einzelnen Studios verwendet die `StudioCard.svelte`. Über den Button «Add New Studio» gelangt man zur Seite `/studios/create`. Mit einem Klick auf den Studios landet man auf die Detailseite der entsprechenden Studio.

Dateien:

- `lib/components/StudioCard.svelte`
- `routes/studios/+page.svelte`
- `routes/studios/+page.server.js`

Studios Create

Route: `/src/routes/studios/create`



The screenshot shows a web browser window with the URL `https://anibuddy-1.netlify.app/studios/create`. The page has a dark theme with a pink header bar containing the 'ANIBUDDY' logo and navigation links for 'ANIMES', 'GENRES', 'STUDIOS', and 'DASHBOARD'. The main content area is titled 'Create New Studio' and features a form with the following fields:

- Studio Name**: A text input field.
- Founded Year**: A text input field.
- Logo URL**: A text input field containing the placeholder `/images/placeholder.jpg`.
- Description**: A large text area for a detailed description.

At the bottom of the form are two buttons: a pink 'CANCEL' button and a blue 'CREATE STUDIO' button.

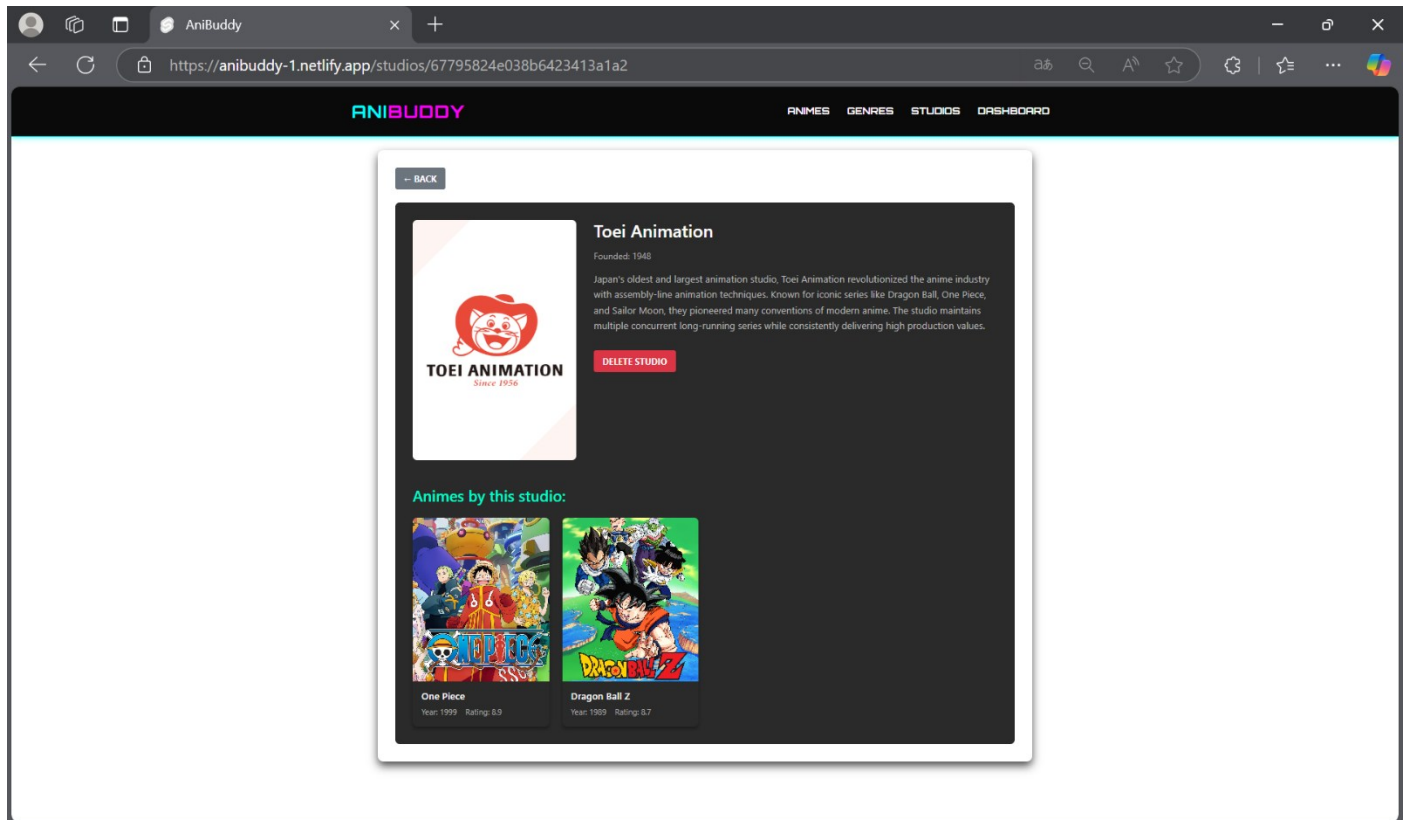
Auf der Seite «Add New Studio» können neue Studios erstellt werden. Die Informationen für das Studio werden in die entsprechenden Felder eingegeben. Mit dem Button «CANCEL» kann das Formular verlassen werden, während der Button «CREATE STUDIO» das Studio erstellt. Nach dem Erstellen gelangt man zum neuen Objekt.

Dateien:

- `routes/studios/create/+page.svelte`
- `routes/studios/create/+page.server.js`

Studios Detailseite

Route: `/src/routes/studios/[studio_id]`



Auf der Detailseite sieht man ein Bild des Studios sowie die dazugehörigen Informationen. Mit dem Button «BACK» gelangt man zurück zur Seite `/studios`. Der Button «DELETE STUDIO» löscht das Studio und führt ebenfalls zurück zur Seite `/studios`. Die Animes des entsprechenden Studios werden angezeigt und sind klickbar

Dateien:

- `routes/animes/[studio_id]/+ page.svelte`
- `routes/animes/[studio_id]/+page.server.js`

Erweiterungen

Watchlist-Filter

Auf der Seite /animes können Benutzer Animes, die nicht auf ihrer Watchlist stehen, mittels einer Checkbox ausblenden. Dies wurde durch die Implementierung eines Formulars und einer clientseitigen Filterlogik in der Datei +page.svelte realisiert. Die Filterung erfolgt dynamisch, indem die Animes basierend auf dem watchlist-Attribut gefiltert werden. Wenn die Checkbox aktiviert ist, werden nur die Animes angezeigt, die auf der Watchlist stehen, was den Benutzern eine personalisierte Ansicht ihrer bevorzugten Animes ermöglicht.

Dateien:

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

User Rating

Auf der Seite /animes/[anime_id] können Benutzer Animes bewerten und Kommentare hinterlassen. Dies wurde mit einem Formular und einer serverseitigen Funktion addRating in der Datei db.js realisiert. Die Bewertungen werden in der MongoDB-Datenbank gespeichert und auf der Seite angezeigt.

Dateien:

- lib/server/db.js
- routes/animes/+page.svelte
- routes/animes/+page.server.js

Erweiterte Datenbankabfragen

Die Anwendung verwendet erweiterte Datenbankabfragen, um Animes mit ihren zugehörigen Genres und Studios zu laden. Dies wurde durch die Implementierung von Aggregations-Pipelines in MongoDB realisiert. Diese Pipelines ermöglichen es, komplexe Abfragen durchzuführen, die mehrere Collections kombinieren und die Daten in einem einzigen Schritt abrufen. Dadurch wird die Effizienz der Datenbankabfragen erhöht und die Daten können direkt in der gewünschten Struktur an die Anwendung übergeben werden.

Dateien:

- /lib/server/db.js

Erstellung und Verwaltung von Genres und Studios

Auf den Seiten /genres/create und /studios/create können Benutzer neue Genres und Studios erstellen und verwalten. Dies wurde durch die Implementierung von Formularen und serverseitigen Funktionen in der Datei db.js realisiert. Die Formulare ermöglichen es den Benutzern, die erforderlichen Informationen einzugeben, die dann in der MongoDB-Datenbank gespeichert werden. Die serverseitigen Funktionen sorgen dafür, dass die Daten korrekt verarbeitet und in die entsprechenden Collections eingefügt werden. Diese Erweiterung erleichtert die Verwaltung und Organisation der Genres und Studios innerhalb der Anwendung.

Dateien:

- /lib/server/db.js
- /routes/genres/[id]/+page.svelte
- /routes/genres/[id]/+page.server.js
- /routes/studios/[id]/+page.svelte
- /routes/studios/[id]/+page.server.js

Dynamische Navigation und Layout

Die Anwendung verfügt über eine dynamische Navigation und ein Layout, das mit SvelteKit implementiert wurde. Dies ermöglicht eine benutzerfreundliche Navigation durch die verschiedenen Seiten der Anwendung. Die Navigation und das Layout passen sich dynamisch an die Inhalte und den Zustand der Anwendung an, was eine nahtlose Benutzererfahrung gewährleistet.

Dateien:

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

Dashboard

Die Funktion `load()` lädt parallel Animes, Genres, Studios und Benutzerbewertungen über die Datenbankmethoden `db.getAnimes()`, `db.getGenres()`, `db.getStudios()` und `db.getRatings()`. Anschliessend berechnet die Anwendung wichtige Kennzahlen, wie die durchschnittliche Bewertung aller Animes (`avgRating`) durch die Summierung aller Bewertungen und Division durch die Anzahl der Animes. Ausserdem erstellt die Funktion eine Übersicht der Studio-Statistiken, indem die Anzahl der Animes pro Studio gezählt und sortiert wird (`studioStats`). Für Jahresstatistiken werden die Veröffentlichungen pro Jahr analysiert und chronologisch geordnet (`yearStats`). Zusätzlich wird die Verteilung der Benutzerbewertungen von 1 bis 5 Sternen in einem Array zusammengefasst (`ratingDistribution`).

Dateien:

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