

Prototyping Projektdokumentation

Name: Daniele Magnano

E-Mail: magnadan@students.zhaw.ch

URL der deployten Anwendung: [https://<appname>.netlify.app/]

1. Einleitung

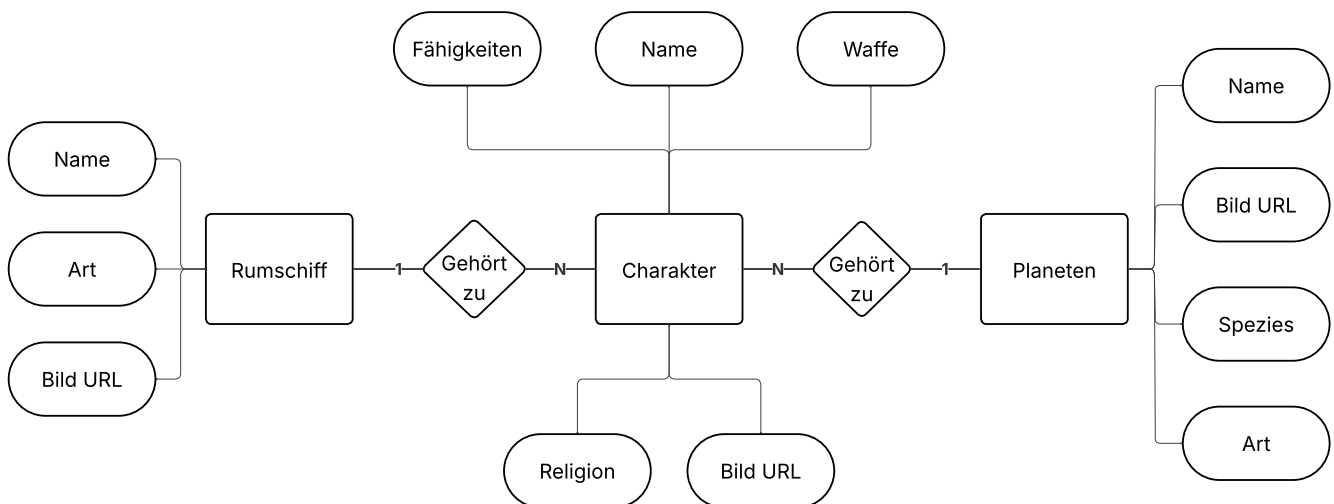
Star Wars Universe ist eine interaktive Webanwendung, die es ermöglicht, zentrale Elemente des Star Wars-Universums wie Charaktere, Planeten und Raumschiffe zu entdecken und zu erweitern. Die Anwendung wurde mit SvelteKit 5 im Runes-Mode umgesetzt und verwendet eine MongoDB-Datenbank zur Verwaltung der Daten.

Die Grundfunktionen der App beinhalten das Erstellen, Anzeigen und Löschen von Charakteren, Planeten und Raumschiffen. Um einen neuen Charakter anlegen zu können, müssen der zugehörige Heimatplanet sowie das passende Raumschiff bereits zuvor erstellt worden sein.

Man kann sich die App als ein stetig wachsendes Star Wars-Lexikon vorstellen, das von einer Community laufend erweitert werden könnte.

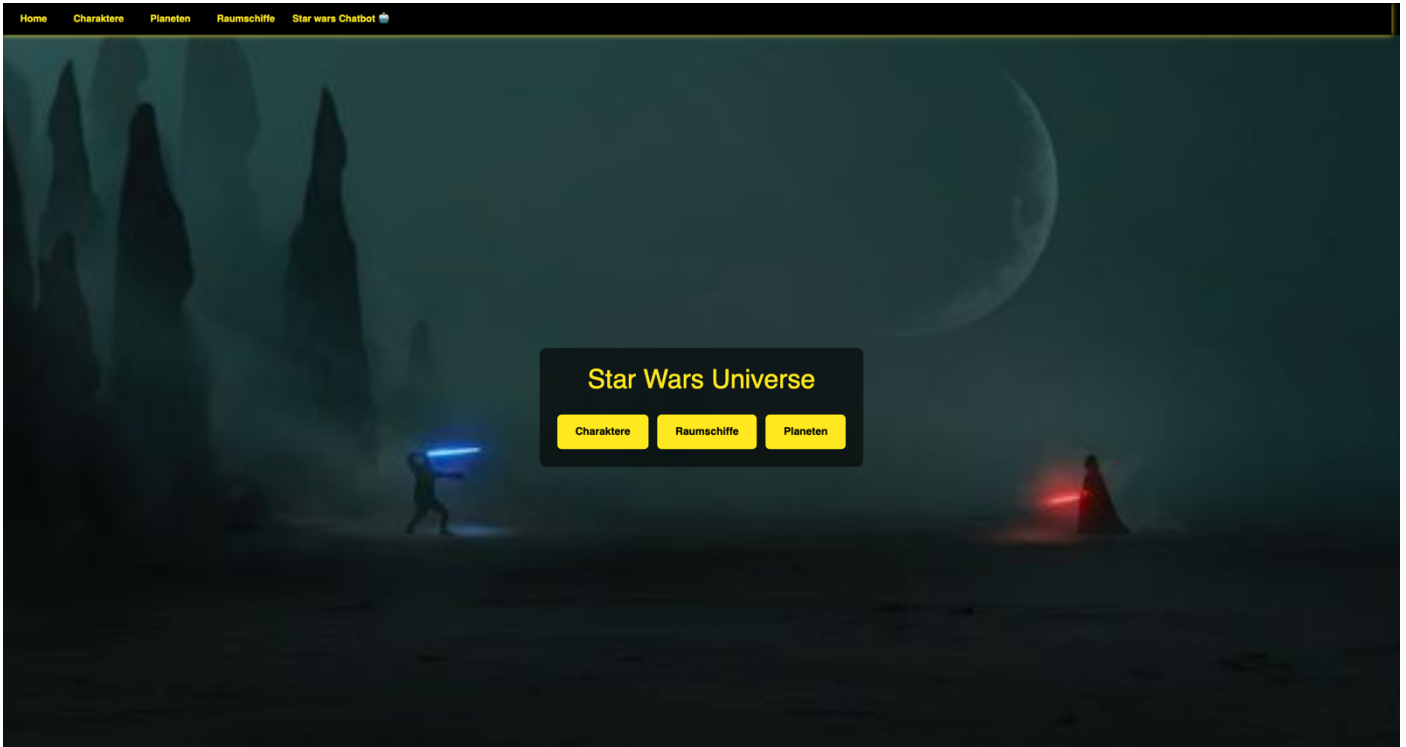
Als zusätzliche Erweiterung wurde ein Chatbot integriert, der über die OpenAI-API mit dem Modell GPT-3.5-Turbo kommuniziert. Nutzer:innen können dem Chatbot Fragen zum Star Wars-Universum stellen und erhalten in Echtzeit passende Antworten, was die Anwendung um eine unterhaltsame und interaktive Komponente ergänzt.

2. Datenmodell



3. Beschreibung der Anwendung

3.1. Startseite



Route: /

Die Startseite dient als zentrale Navigation und Einstiegspunkt in die Anwendung. Im Mittelpunkt steht ein hero-artiges Hintergrundbild, das eine ikonische Szene aus dem Star Wars-Universum zeigt (Anakin vs. Obi-Wan). Darüber liegt ein transparenter Block mit dem Titel „**Star Wars Universe**“ sowie drei Buttons, die direkt zu den Bereichen **Charaktere**, **Raumschiffe** und **Planeten** führen.

Zusätzlich befindet sich im oberen Bereich eine feststehende Navigationsleiste, über die man ebenfalls zwischen den Hauptbereichen und dem Chatbot navigieren kann. Die Navigation bleibt beim Scrollen sichtbar und ist visuell durch einen schwarzen Hintergrund mit gelber Schrift im Star Wars-Stil abgesetzt.

Dateien:

- routes/+page.svelte
- routes/+layout.svelte (Navigation)

3.2 Charaktere

The screenshot shows a web application interface for creating and managing Star Wars characters. At the top, a navigation bar includes links for Home, Charaktere, Planeten, Raumschiffe, and Star wars Chatbot. The main section is titled 'Charakter erstellen' and contains a form with the following fields: Name (text input), Religion (text input), Waffe (text input), Fähigkeiten (Komma getrennt) (text input), Bild-URL (text input), Planet (dropdown menu with 'Bitte wählen' selected), and Raumschiff (dropdown menu with 'Bitte wählen' selected). A yellow button labeled 'Charakter speichern' is at the bottom of the form. Below the form, the 'Charaktere' section displays a list of characters in a card view. The first card shows 'Luke Skywalker' with details: Religion: Jedi, Waffe: Grünes Lichtschwert, Fähigkeiten: Machtelekinese, Lichtschwertkampf, Gedankenbeeinflussung, Machtteilung, taktisches Denken, Planet: Tatooine, and Raumschiff: X-Wing. A red 'Löschen' button is at the bottom of the card. To the right of the card are three character images: Darth Vader, Han Solo, and Chewbacca.

Route: /characters

Auf dieser Seite können neue Star Wars-Charaktere erstellt und bestehende angezeigt bzw. gelöscht werden. Das Formular zur Charaktererstellung erfasst Name, Religion, Waffe, Fähigkeiten (kommasetrennt), Bild-URL sowie den zugehörigen Heimatplaneten und das Raumschiff. Damit ein Charakter gespeichert werden kann, müssen sowohl ein gültiger Planet als auch ein Raumschiff aus den bestehenden Einträgen ausgewählt werden.

Unterhalb des Formulars werden alle gespeicherten Charaktere in einer Kartenansicht dargestellt. Die Karten zeigen jeweils ein Bild sowie zentrale Informationen wie Religion, Waffe, Fähigkeiten, Heimatplanet und Raumschiff. Jeder Charakter kann über den Roten Button auf der Rückseite der Karte direkt gelöscht werden.

Dateien:

- routes/characters/+page.svelte
- routes/characters/+page.server.js
- lib/db.js (Funktionen createCharacter, getCharacters, deleteCharacter)

3.3 Planeten

[Home](#) [Charaktere](#) [Planeten](#) [Raumschiffe](#) [Star wars Chatbot](#)

Planet erstellen

Name:


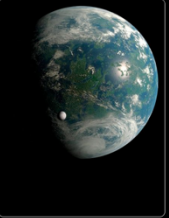


Spezies:

Art:

Bild-URL:

Planet speichern

Gespeicherte Planeten



Route: /planets

Auf dieser Seite können neue Planeten erfasst und bestehende angezeigt werden. Das Formular zur Erstellung eines Planeten umfasst die Felder **Name**, **Art**, **Spezies** und **Bild-URL**. Nach dem Speichern wird der neue Planet in der MongoDB-Datenbank abgelegt und erscheint automatisch in der Übersicht.

Im unteren Bereich der Seite werden alle gespeicherten Planeten als Karten mit Bild dargestellt. Die Kartenansicht dient der Übersicht und stellt die Planeten visuell in den Mittelpunkt. Die erstellten Planeten können später beim Erstellen von Charakteren als Heimatplanet ausgewählt werden. Die Planeten können ebenfalls wieder auf der Rückseite der Karte über roten Button gelöscht werden.

Dateien:

- routes/planets/+page.svelte
- routes/planets/+page.server.js
- lib/db.js (Funktionen createPlanet, getPlanets, deletePlanet)

3.4 Raumschiffe

[Home](#) [Charaktere](#) [Planeten](#) [Raumschiffe](#) [Star wars Chatbot](#)

Raumschiff erstellen





Name:

Art:

Bild-URL:

Raumschiff speichern

Gespeicherte Raumschiffe



Route: /spaceships

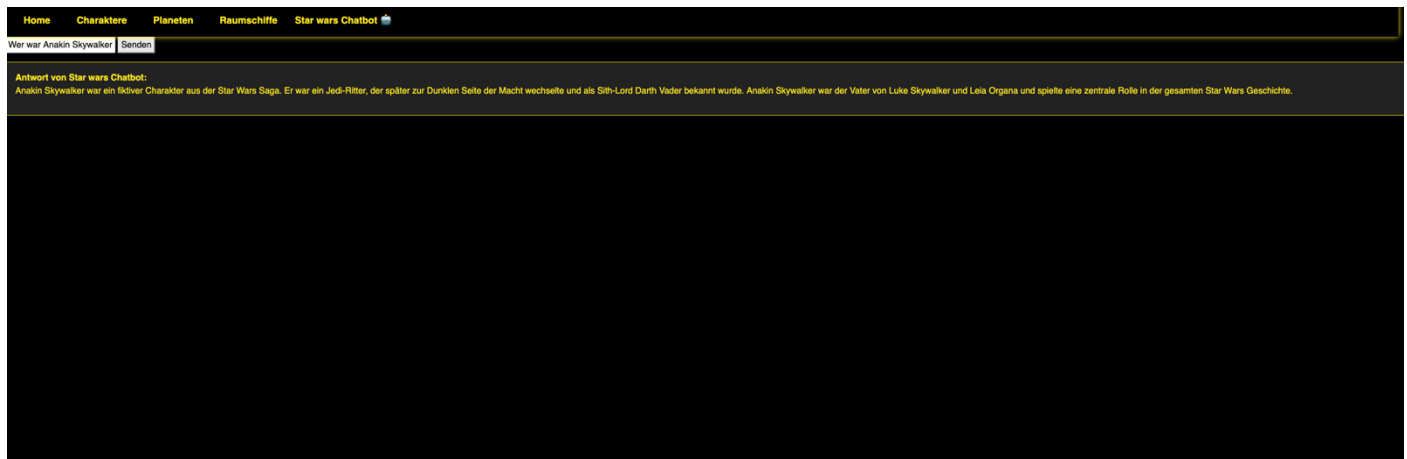
Auf dieser Seite können neue Raumschiffe hinzugefügt und bestehende angezeigt werden. Das Formular zur Erstellung eines Raumschiffs umfasst die Felder **Name**, **Art** und **Bild-URL**. Nach dem Speichern wird das Raumschiff in der Datenbank abgelegt und direkt in der Übersicht dargestellt.

Im unteren Bereich der Seite werden alle gespeicherten Raumschiffe als Bildkarten angezeigt. Die Raumschiffe dienen als referenzierte Einträge und können später bei der Erstellung eines Charakters einem Charakter zugewiesen werden. Die Raumschiffe können ebenfalls wieder auf der Rückseite der Karte über roten Button gelöscht werden.

Dateien:

- routes/spaceships/+page.svelte
- routes/spaceships/+page.server.js
- lib/db.js (Funktionen createSpaceship, getSpaceships, deleteSpaceships)

3.5 Chatbot



Route: /chat

Auf dieser Seite wurde ein interaktiver Chatbot integriert, der Fragen zum Star Wars-Universum in Echtzeit beantwortet. Nutzer:innen können eine beliebige Frage in das Eingabefeld eingeben und erhalten unmittelbar eine KI-generierte Antwort. Die Kommunikation erfolgt über die OpenAI-API mit dem Modell gpt-3.5-turbo. Die Chats werden nicht gespeichert und gehen bei einer Seitenneuladung oder nach einer neuen Frage verloren.

Die Seite ist bewusst minimalistisch gehalten, um die Frage-Antwort-Interaktion in den Fokus zu stellen. Die Eingabe erfolgt über ein Textfeld mit Button, die Antwort wird direkt darunter angezeigt. Der Chatbot erkennt kontextbezogene Fragen wie z. B. „Wer war Anakin Skywalker?“ und liefert passende Informationen in natürlicher Sprache.

Dateien:

- routes/chat/+page.svelte
- routes/chat/api/+server.js

4. Erweiterungen

4.1. Chatbot-Integration

Auf der Seite /chat wurde ein interaktiver Chatbot implementiert, der Fragen zum Star Wars-Universum in Echtzeit beantwortet. Die Kommunikation erfolgt über die OpenAI-API mit dem Modell gpt-3.5-turbo. Nutzer:innen können beliebige Fragen stellen (z. B. „Wer ist Yoda?“ oder „Was ist der Todesstern?“), woraufhin eine direkte, kontextbezogene Antwort vom KI-Modell generiert und angezeigt wird.

Technisch wurde dies durch eine benutzerdefinierte GET-API-Route umgesetzt, die serverseitig mit +server.js realisiert ist. Die Benutzeroberfläche nutzt fetch() für asynchrone Datenabfragen und reagiert ohne Seitenreloads. Diese Erweiterung erweitern die Anwendung um eine unterhaltsame, dynamische und informative Komponente.

Datei(en):

- routes/chat/+page.svelte
- routes/chat/api/+server.js

4.2. Verknüpfung zwischen Charakteren, Planeten und Raumschiffen

Beim Erstellen eines Charakters müssen ein existierender Heimatplanet und ein Raumschiff ausgewählt werden. Diese Referenzen werden gespeichert und in der Charakter-Übersicht sowie in der Detailansicht korrekt dargestellt. Dadurch entsteht eine relationale Datenstruktur mit klaren Abhängigkeiten zwischen den Entitäten.

Datei(en):

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