Prototyping Projektdokumentation

Name: Laura Stärk

E-Mail: staerlau@students.zhaw.ch

URL der deployten Anwendung: vault-of-legends.netlify.app

1. Einleitung

Vault of Legends ist eine Webanwendung zur Verwaltung eigener Dungeons & Dragons-Charaktere. Nutzer:innen können neue Charaktere erstellen, bearbeiten und anzeigen – inklusive zentraler Spielwerte wie Klassen, Spezies, Fähigkeiten, Rüstungsklasse und Lebenspunkte.

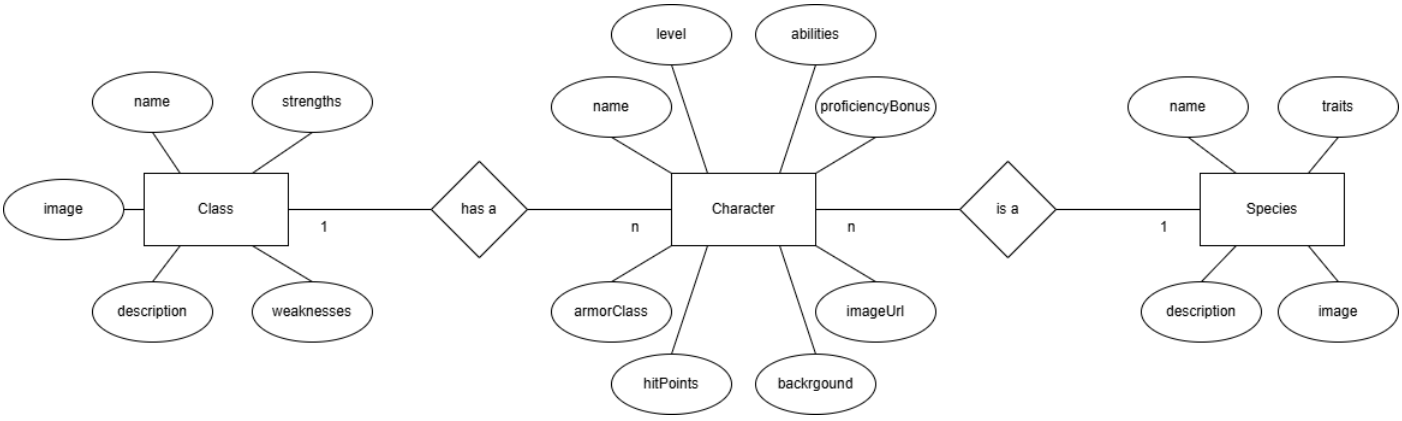
Die Anwendung richtet sich an Spieler:innen, die ihre DnD-Charaktere digital organisieren möchten, ohne sich durch das gesamte Regelwerk arbeiten zu müssen.

Grundfunktionen:

- Charakter anlegen mit Name, Klasse, Spezies, Level, Werten und Bild
- Charakter bearbeiten (z. B. Level-Up, neue Fähigkeiten eintragen)
- Übersicht & Detailansicht aller Charaktere
- Listen & Detailseiten für Klassen und Spezies
- Datenhaltung in MongoDB, umgesetzt mit SvelteKit 5 (Runes Mode)

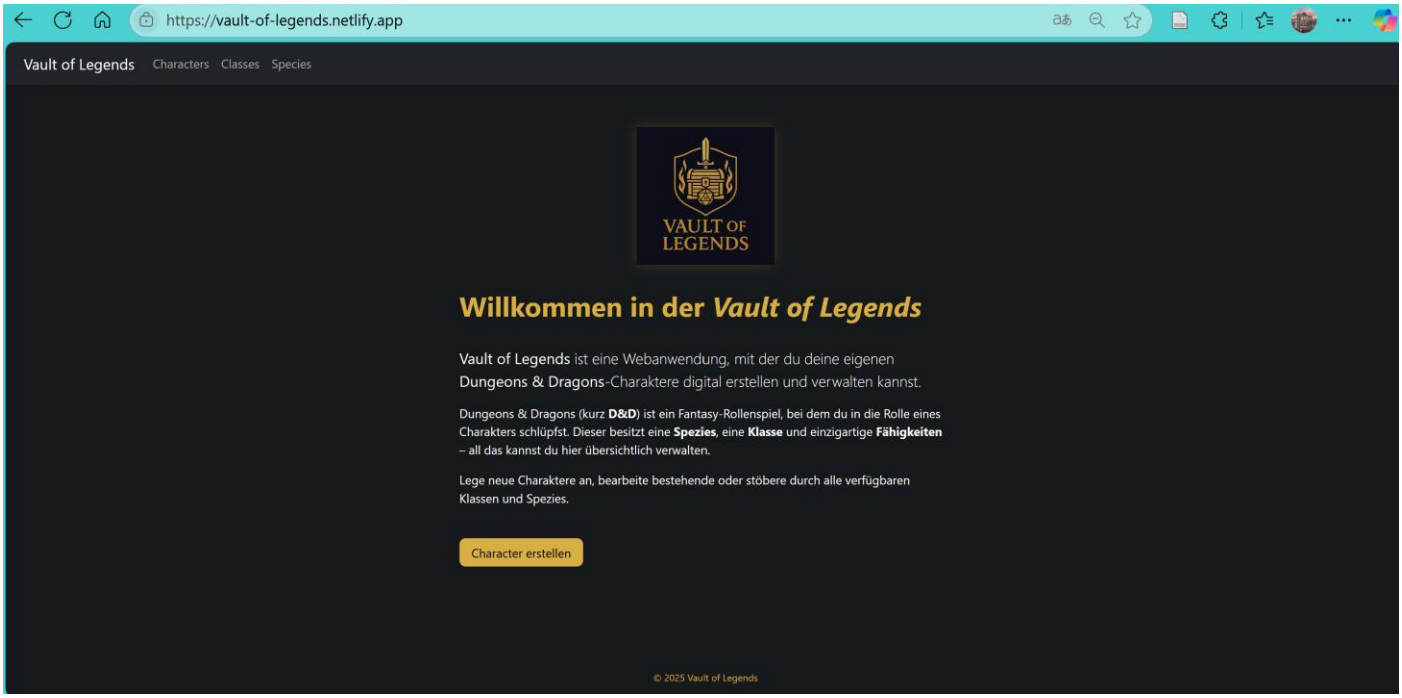
Hinweis: Aufgrund des Umfangs und der Komplexität von D&D können im Rahmen dieses Projekts nicht alle charakterrelevanten Informationen vollständig auf der Website abgebildet werden – insbesondere regelbasierte Werte wie Skills, die berechnet werden müssen. Aus denselben Gründen gibt es keine Erklärungen spezifischer D&D-Begriffe. Die Sprache ist bewusst zweisprachig gehalten, da viele Fachbegriffe im D&D-Kontext auf Englisch üblich sind.

2. Datenmodell



3. Beschreibung der Anwendung

3.1. Startseite



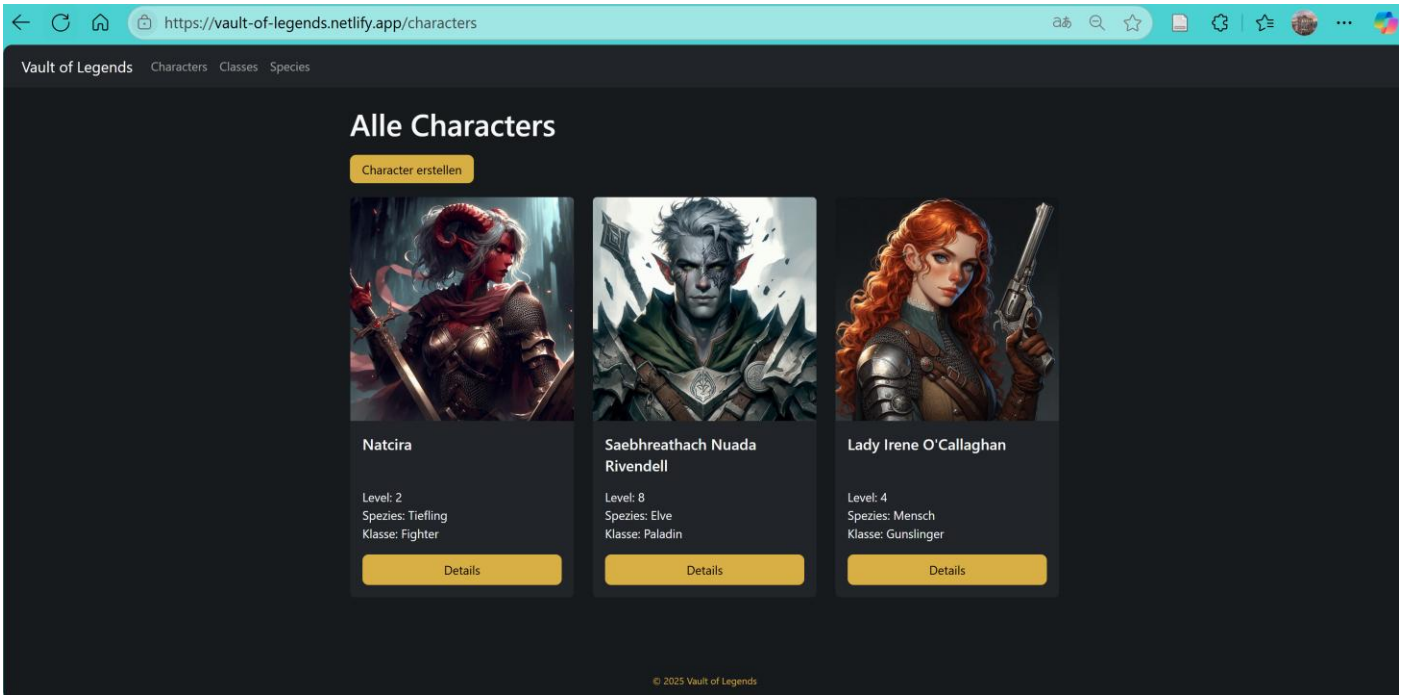
Route: /

Diese Seite begrüsst die Benutzer:innen mit einem Logo und einer kurzen Einführung in Dungeons & Dragons sowie den Funktionsumfang der App. Ein Button führt zur Charaktererstellung.

Dateien:

- routes/+page.svelte
- routes/+page.server.js
- routes/app.html
- lib/components/Navbar.svelte

3.2. Charakterübersicht



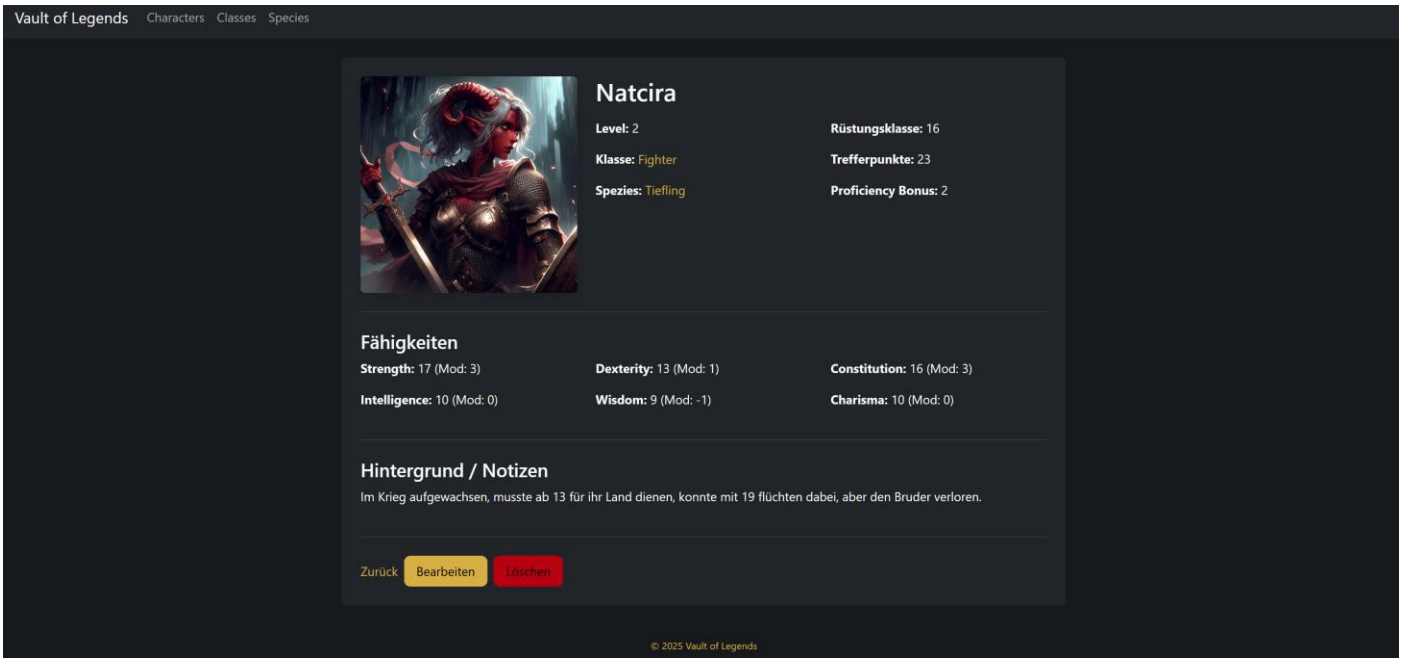
Route: /characters

Mittels Klick auf «Characters» im Navigator gelangt man auf die Charakterübersicht. Diese zeigt alle gespeicherten Charaktere als Karten mit Bild, Name, Level, Klasse, Spezies und einen Button der zur Charakterdetailansicht führt. Weiter gibt es einen Button zur Charaktererstellung.

Dateien:

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

3.3. Charakterdetailansicht



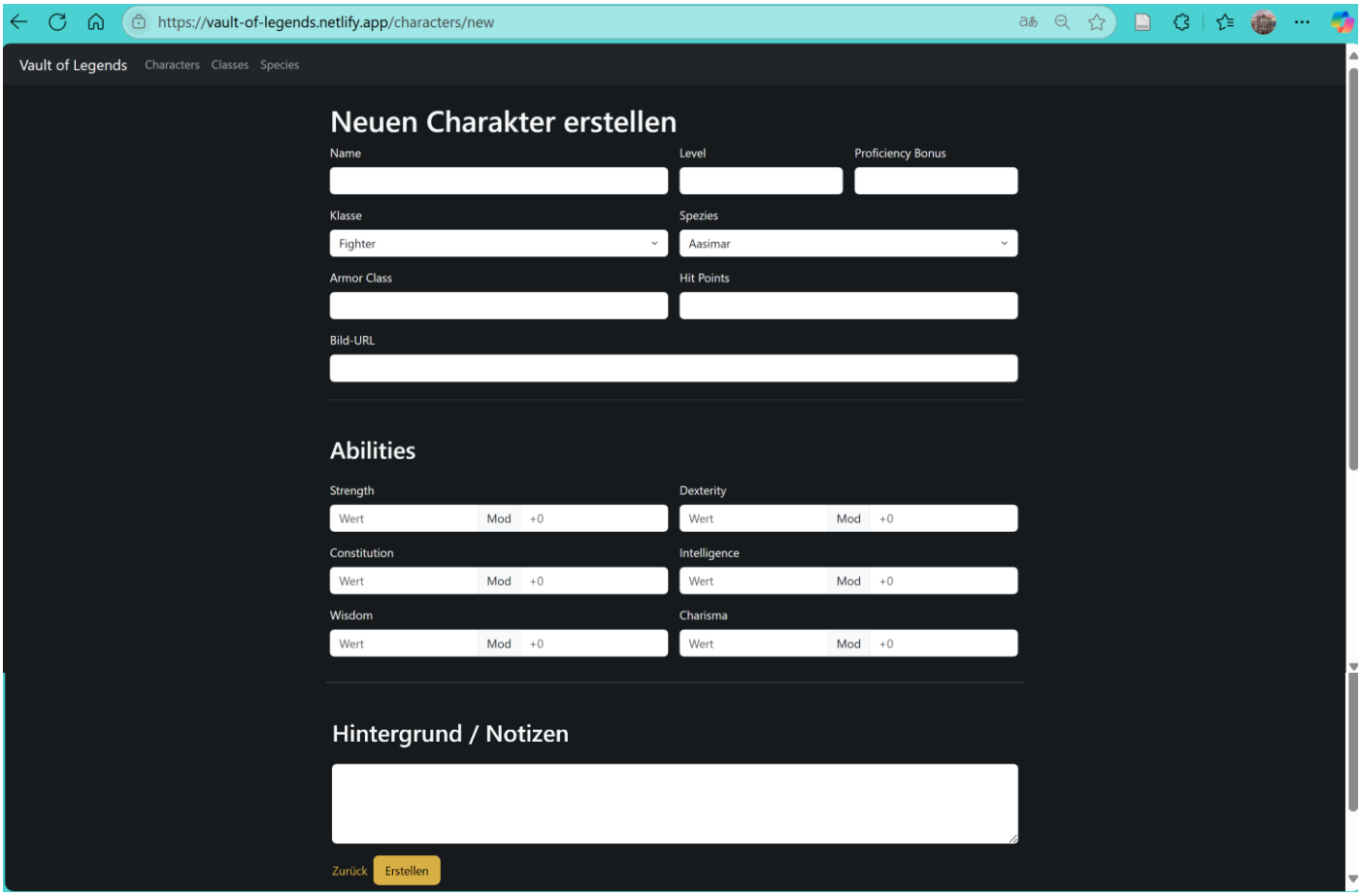
Route: /characters/[id]

Über den Button «Details» auf der Charakterübersicht gelangt man zur Charakterdetailansicht. In dieser Ansicht werden alle Informationen eines Charakters übersichtlich in einer Detailkarte angezeigt: Bild, Level, Klasse, Spezies, Rüstungsklasse, Trefferpunkte, Proficiency Bonus, Fähigkeiten und Hintergrundtext. Zusätzlich gibt es Buttons zum Bearbeiten und Löschen, sowie um zurück zur Charakterübersicht zu gelangen. Beim Klick auf Klassen- oder Speziesnamen gelangt man direkt zur zugehörigen Detailansicht.

Dateien:

- routes/characters/[id]/+page.svelte
- routes/characters/[id]/+page.server.js
- lib/components/CharacterDetailCard.svelte
- lib/server/db.js

3.4. Charakter erstellen



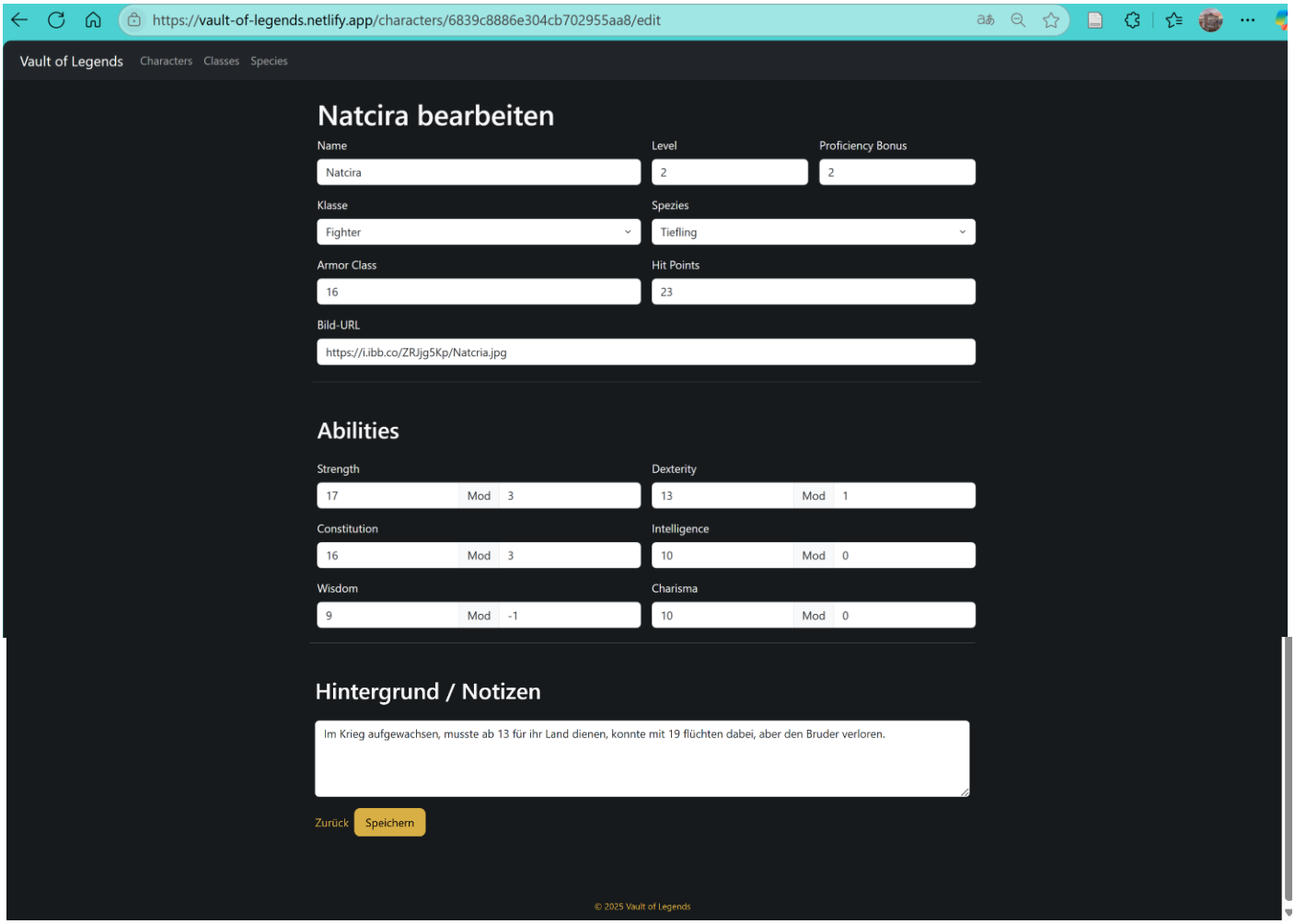
Route: /characters/new

Klickt man von der Startseite oder von der Charakterübersicht auf «Character erstellen», landet man bei einem strukturierten Formular, mit welchem ein neuer D&D-Charakter angelegt werden kann. Felder sind z. B. Name, Klasse, Spezies, Level, Fähigkeiten, Trefferpunkte und Bild. Wird keine Bild-URL angegeben, wird automatisch ein Platzhalter verwendet. Durch den Button «Erstellen» wird der Charakter gespeichert und auf der Charakterübersicht angezeigt. Durch «Zurück» bricht man den Vorgang ab und landet ebenfalls wieder bei der Charakterübersicht.

Dateien:

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

3.5. Charakter bearbeiten



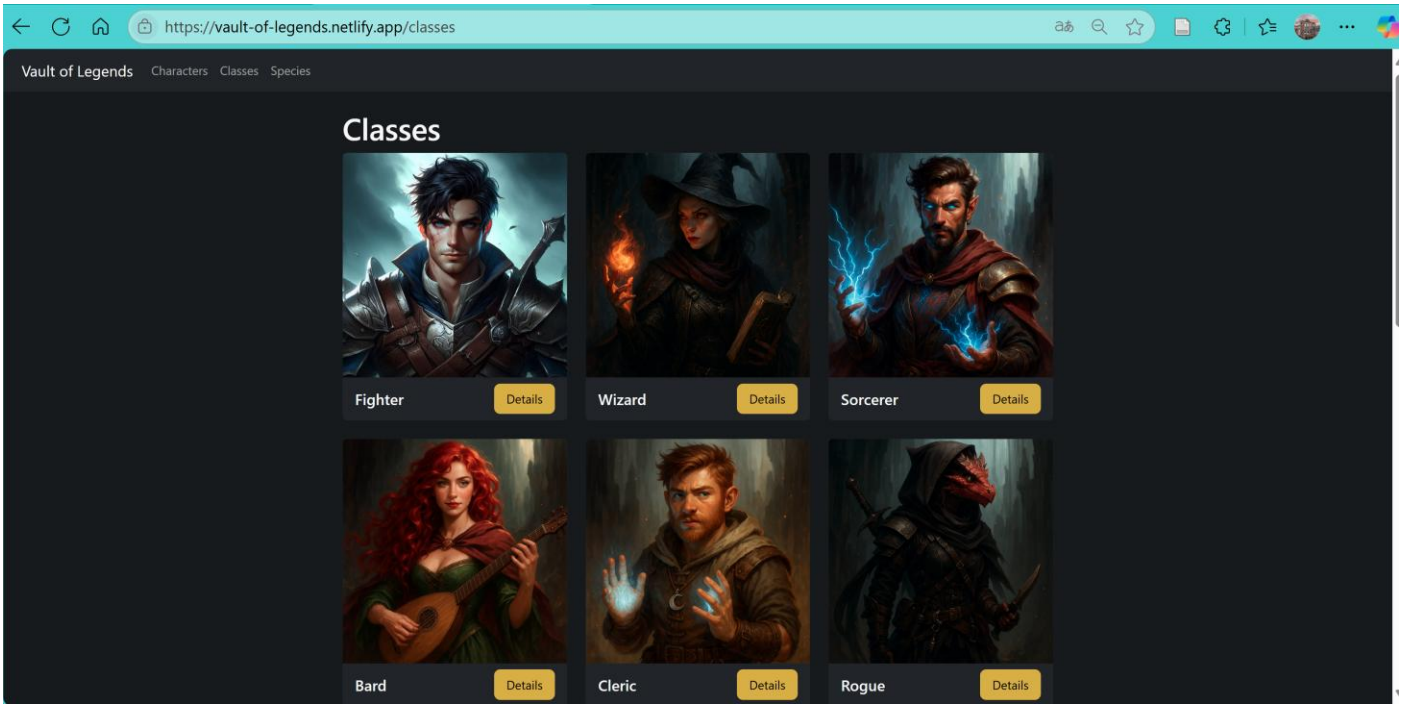
Route: /characters/[id]/edit

Klickt man bei der Charakterdetailansicht auf den Button «Bearbeiten», gelangt man zu einem Formular, welches gleich aussieht wie bei «Character erstellen» – allerdings inkl. der bereits hinterlegten Daten. Die Daten können beliebig geändert werden. Mit einem Klick auf «Speichern» gelangt man wieder bei der Charakterdetailansicht mit den angepassten Daten. Mit einem klick auf «Zurück» wird der Vorgang abgebrochen.

Dateien:

- routes/characters/[id]/edit/+page.svelte
- routes/characters/[id]/edit/+page.server.js
- lib/components/CharacterForm.svelte
- lib/server/db.js

3.6. Klassenübersicht



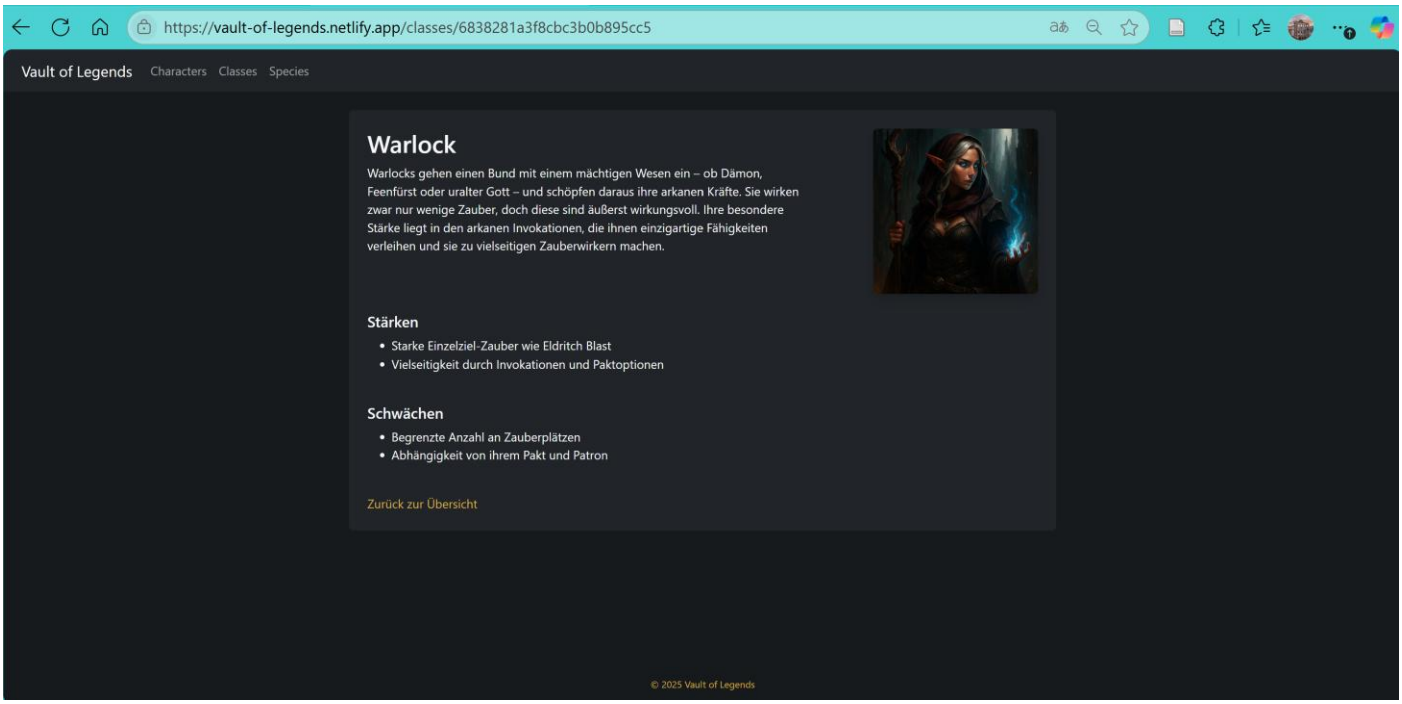
Route: /classes

Mittels Klick auf «Classes» im Navigator gelangt man auf die Klassenübersicht. In der Übersicht werden alle D&D-Klassen als Bilder-Cards angezeigt – inkl. Namen, Bild und Button, der zur Detailansicht führt.

Dateien:

- routes/classes/+page.svelte
- routes/classes/+page.server.js
- lib/components/ClassCard.svelte
- lib/server/db.js

3.7. Klassendetailansicht



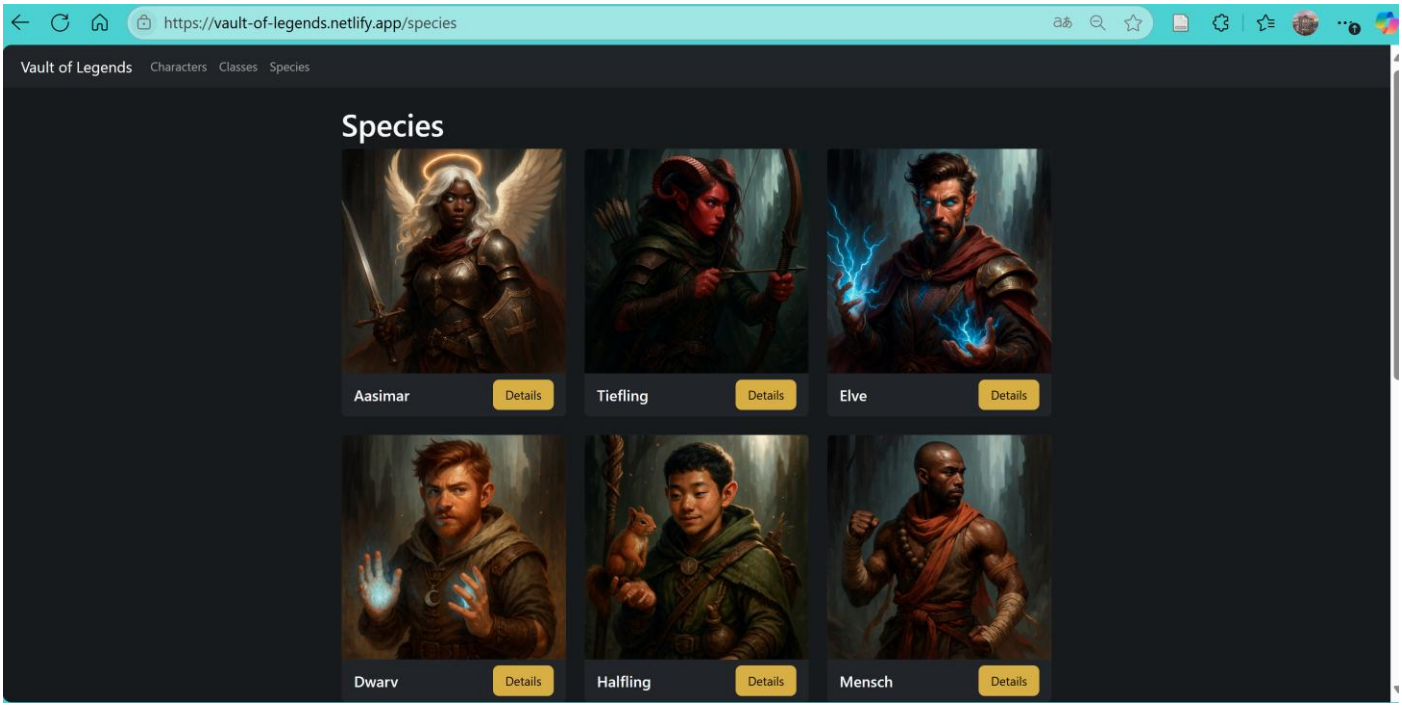
Route: /classes/[id]

Mittels Klick auf «Details» auf der Klassenübersicht gelangt man zur jeweiligen Klassendetailansicht. In der Detailansicht werden Beschreibung, Stärken, Schwächen und ein Klassenbild dargestellt sowie ein Button, der zur vorherigen Seite zurückführt.

Dateien:

- routes/classes/[id]/+page.svelte
- routes/classes/[id]/+page.server.js
- lib/components/ClassDetailCard.svelte
- lib/server/db.js

3.8. Speziesübersicht



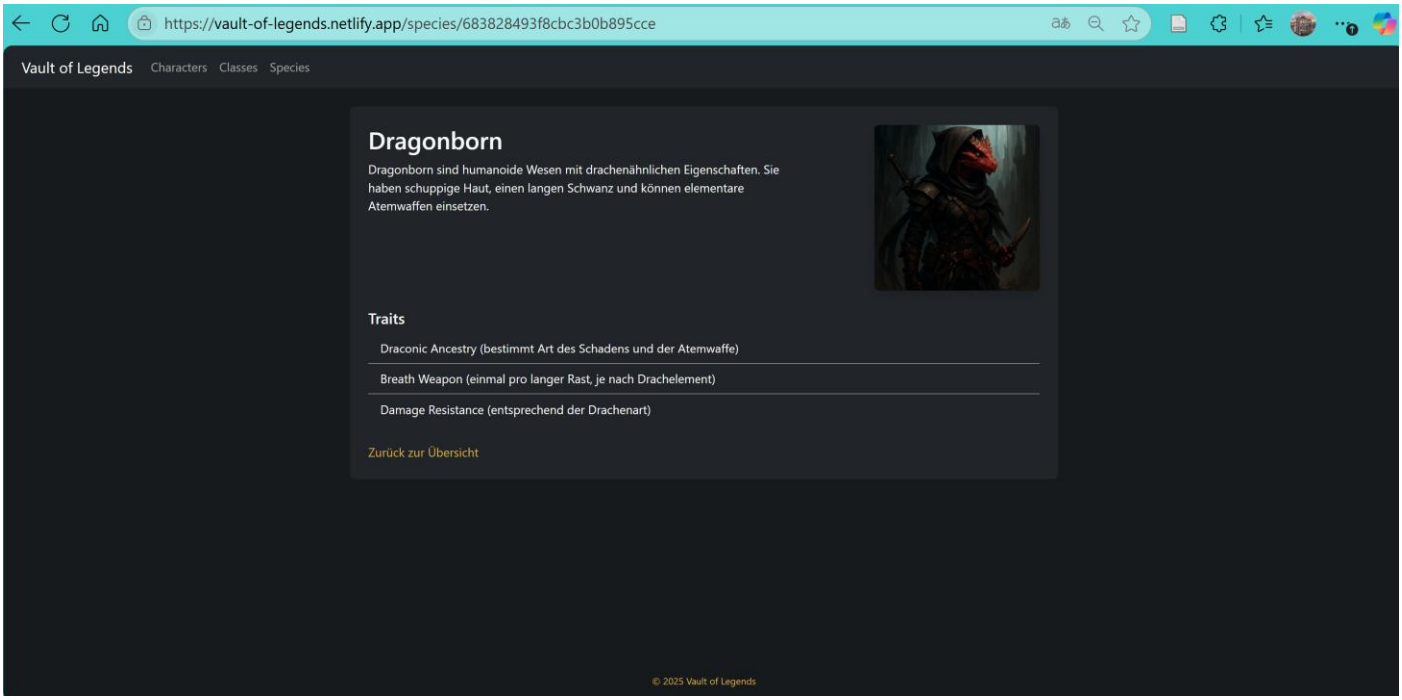
Routen: /species

Mittels Klick auf «Species» im Navigator gelangt man auf die Speziesübersicht. In der Übersicht werden alle D&D-Spezies als Bilder-Cards angezeigt – inkl. Namen, Bild und Button, der zur Detailansicht führt.

Dateien:

- routes/species/+page.svelte
- routes/species/+page.server.js
- lib/components/SpeciesCard.svelte
- lib/server/db.js

3.9. Speziesdetailansicht



Routen: /species/[id]

Mittels Klick auf «Details» auf der Speziesübersicht gelangt man zur jeweiligen Speziesdetailansicht. In der Detailansicht werden Beschreibung, die Traits (Merkmale) und ein Klassenbild dargestellt sowie ein Button, der zur vorherigen Seite zurückführt.

Dateien:

- routes/species/[id]/+page.svelte
- routes/species/[id]/+page.server.js
- lib/components/SpeciesDetailCard.svelte
- lib/server/db.js

4. Erweiterungen

4.1. Zusätzlicher Inhalt

Ergänzend zu den Basisanforderungen wurde eine weitere Page erstellt für den dritten Entitätstyp (/species), zwei weitere Pages mit Detailansichten (/species/[id] und /classes/[id]) sowie eine zweite Page mit Formular zur Dateneingabe bzw. -änderung (/characters/[id]/edit). All diese Funktionen sind in «3. Beschreibung der Anwendung» aufgeführt.

4.2. Zusätzliche Components

Ergänzend zu den 5 unterschiedlichen Svelte-Komponenten der Mindestanforderung sind drei zusätzliche erstellt worden – passend zum dritten Entitätstyp Species (lib/components/SpeciesCard.svelte) und den zusätzlichen Detailansichten(lib/components/SpeciesDetailCard.svelte und lib/components/ClassDetailCard.svelte). Falls diese nicht als Erweiterung gelten, da sie sich sehr ähnlich sind, wurde auch eine Component für den Navigator erstellt: lib/components/Navbar.svelte

4.3. Rückverlinkung zur Charakteransicht

Klickt man in der Charakterdetailansicht auf den Klassen- oder Speziesnamen, gelangt man direkt zur zugehörigen Detailansicht. So gibt es zwei Möglichkeiten, auf eine Klassendetailansicht oder eine Speziesdetailansicht zu gelangen (über die jeweilen Übersichtseiten oder über die Verlinkung in der Charakterdetailansicht). Mit dem Zurück-Button

auf der Klassendetailansicht bzw. Speziesdetailansicht wird man auf die Seite zurück geführt, von der man kam. Dies wird auch mit dem Zurück-Button auf Klassendetailansicht bzw. Speziesdetailansicht entsprechend angezeigt:

- ➔ Wird die Detailseite einer Klasse oder Spezies aus einem Charakter geöffnet, erscheint ein Button «Zurück zum Charakter».
- ➔ Wird die Detailseite einer Klasse oder Spezies aus der Übersichtseite geöffnet, erscheint ein Button «Zurück zur Übersicht».

Dies wurde über die Übergabe eines fromCharacter-Parameters und einer if-else-Verzweigung realisiert.

Dateien:

- lib/components/ClassDetailCard.svelte
- lib/components/SpeciesDetailCard.svelte
- lib/components/CharacterDetailCard.svelte

4.4. Platzhalterbild bei fehlender Bild-URL

Wenn kein Bild für einen Charakter vorhanden ist, wird ein Standardbild (/images/Platzhalter.png) angezeigt. Dies wurde mit einer if-else-Verzweigung realisiert.

Dateien:

- lib/components/CharacterCard.svelte
- lib/components/CharacterDetailCard.svelte
- static/images/Platzhalter.png

4.5. Farbschema, Design & abgerundete Cards

Ein eigenes Farbschema (Darkmode + Gold) wurde verwendet. Cards und Bilder wurden optisch mit Bootstrap und Custom CSS angepasst (abgerundet, Schatten etc.).

Dateien:

- app.css
- lib/components/*.svelte

4.6. Eigenes Logo als Favicon

Als Favicon wurde ein selbst erstelltes Logo hinterlegt.

Dateien:

- app.html
- static/Logo.png

4.7. Navigation & Footer

Ein Footer wurde integriert.

Dateien:

- routes/+layout.svelte