

# Prototyping Projektdokumentation

Name: David Schofield

E-Mail: schofdav@students.zhaw.ch

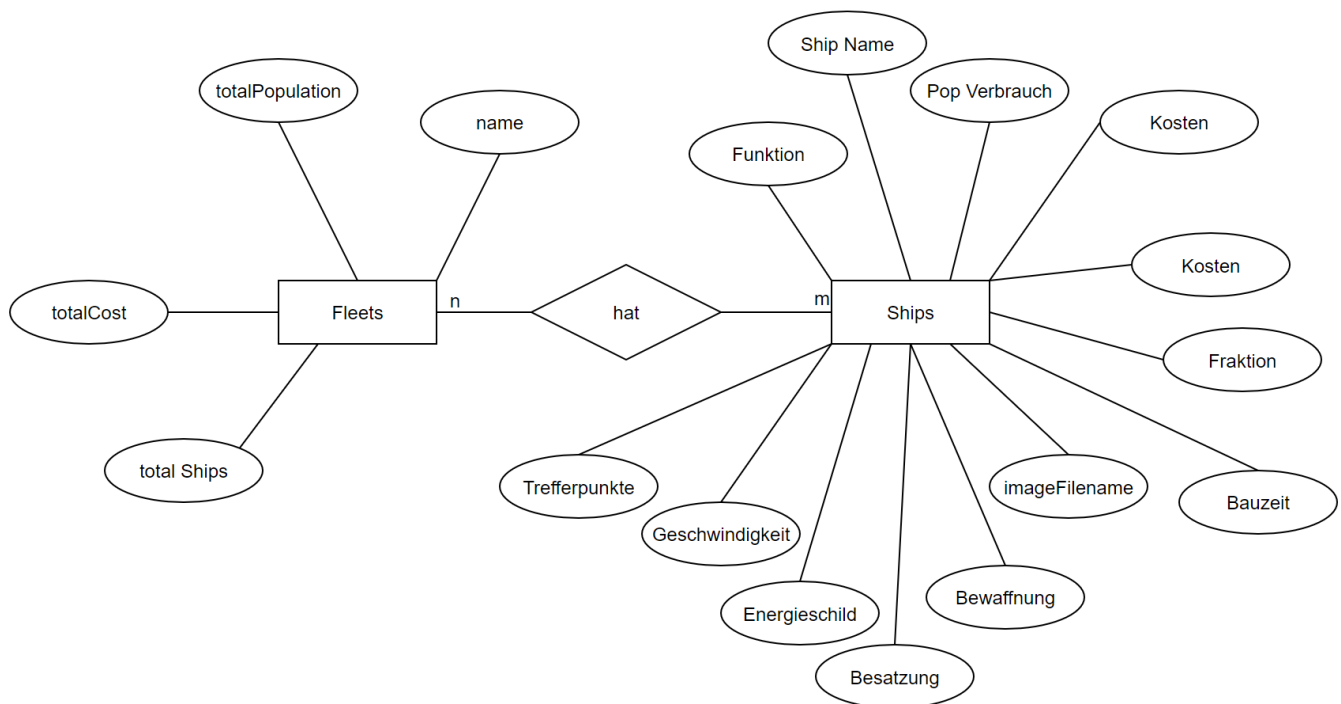
URL der deployten Anwendung: <https://fleetbuildereawx.netlify.app/>

## 1. Einleitung

Empire at War Expanded Thrawn's Revenge ist eine Modifikation für das beliebte Spiel Star Wars Empire at War Forces of Corruption. In diesem Strategiespiel versucht man die Galaxy zu erobern in dem man Flotten baut und Planeten einnimmt. In jedem Kampf dürfen pro Seite Schiffe mit einer addierten Population von 300 gleichzeitig auf dem Schlachtfeld sein.

Mit dem Empire at War Expanded Thrawn's Revenge Fleet Builder kann man Schiffe aus dem Spiel in eine Flotte zusammenbauen und diese miteinander vergleichen. Die Webseite addiert die Kosten der Schiffe und speichert die Flotten, um sie später wieder zu bestaunen. Ebenfalls kann man die einzelnen Schiffstypen genauer unter die Lupe nehmen.

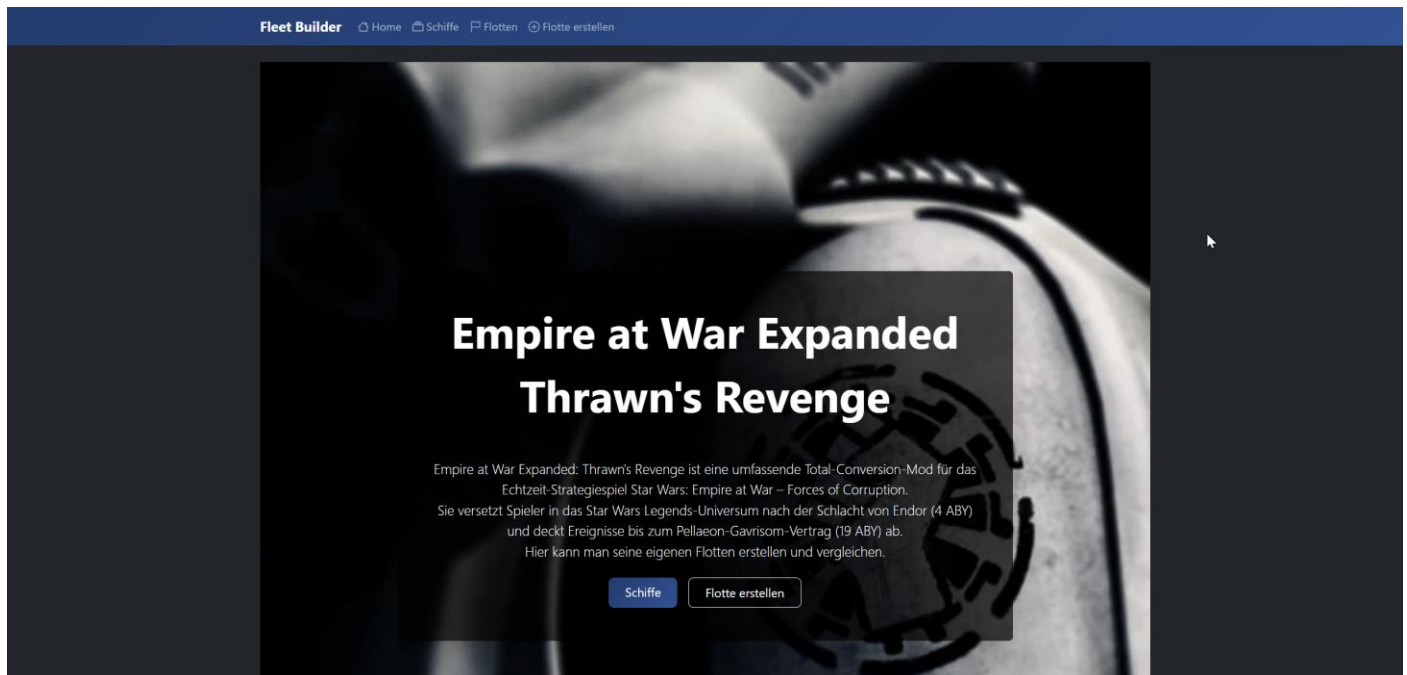
## 2. Datenmodell



## 3. Beschreibung der Anwendung

### 3.1. Homepage

Route: /



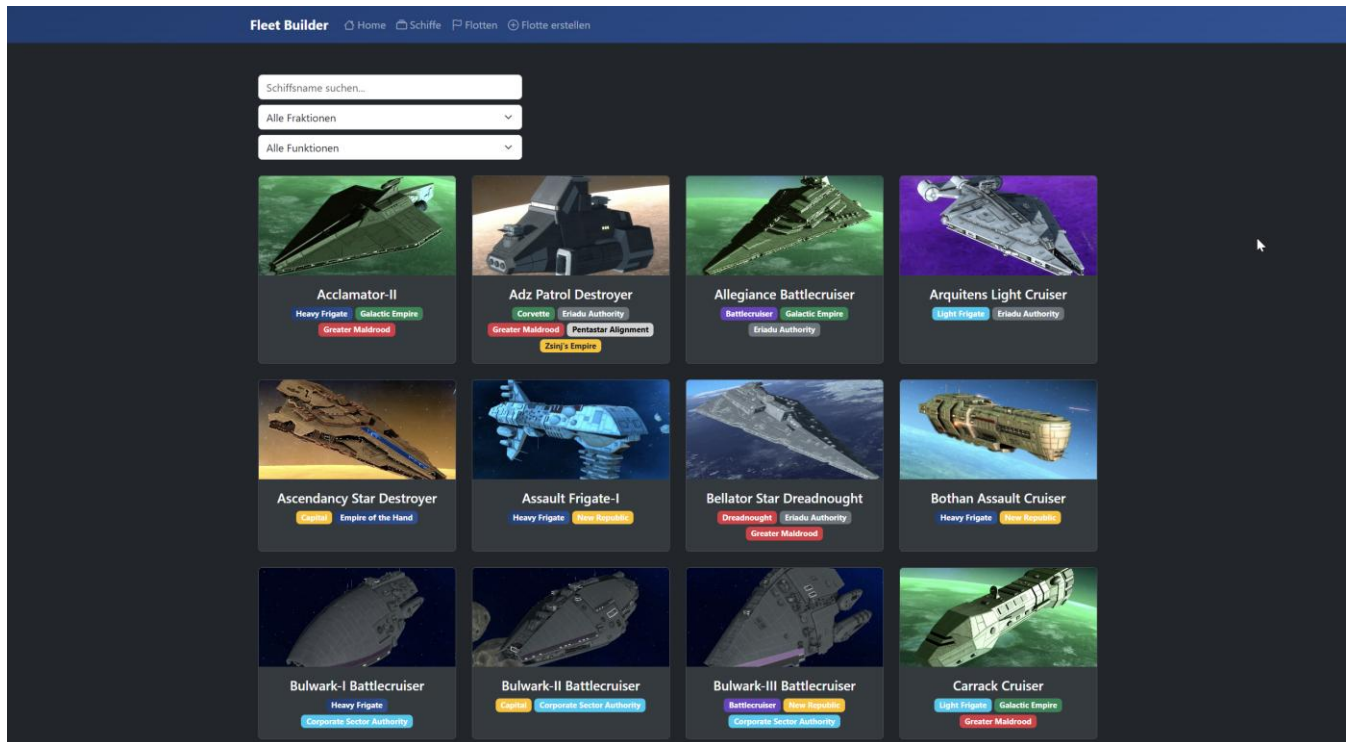
Auf dieser Seite ist eine kurze Einführung ersichtlich und direkte Links zu den Schiffen und zur Flotten Erstellung. Diese sind auch durch die Links oben erreichbar.

Dateien:

- routes/+page.svelte - Homepage-Komponente
- routes/+page.server.js - Lädt Daten für Homepage
- routes/+layout.svelte - Navigation und Layout
- lib/db.js - Datenbankfunktionen (getShips, getFleets, getFactions, getShipTypes)
- lib/db-fallback.js - Fallback-Daten
- app.css - Globale Styles

## 3.2. Schiffe

Route: /ships



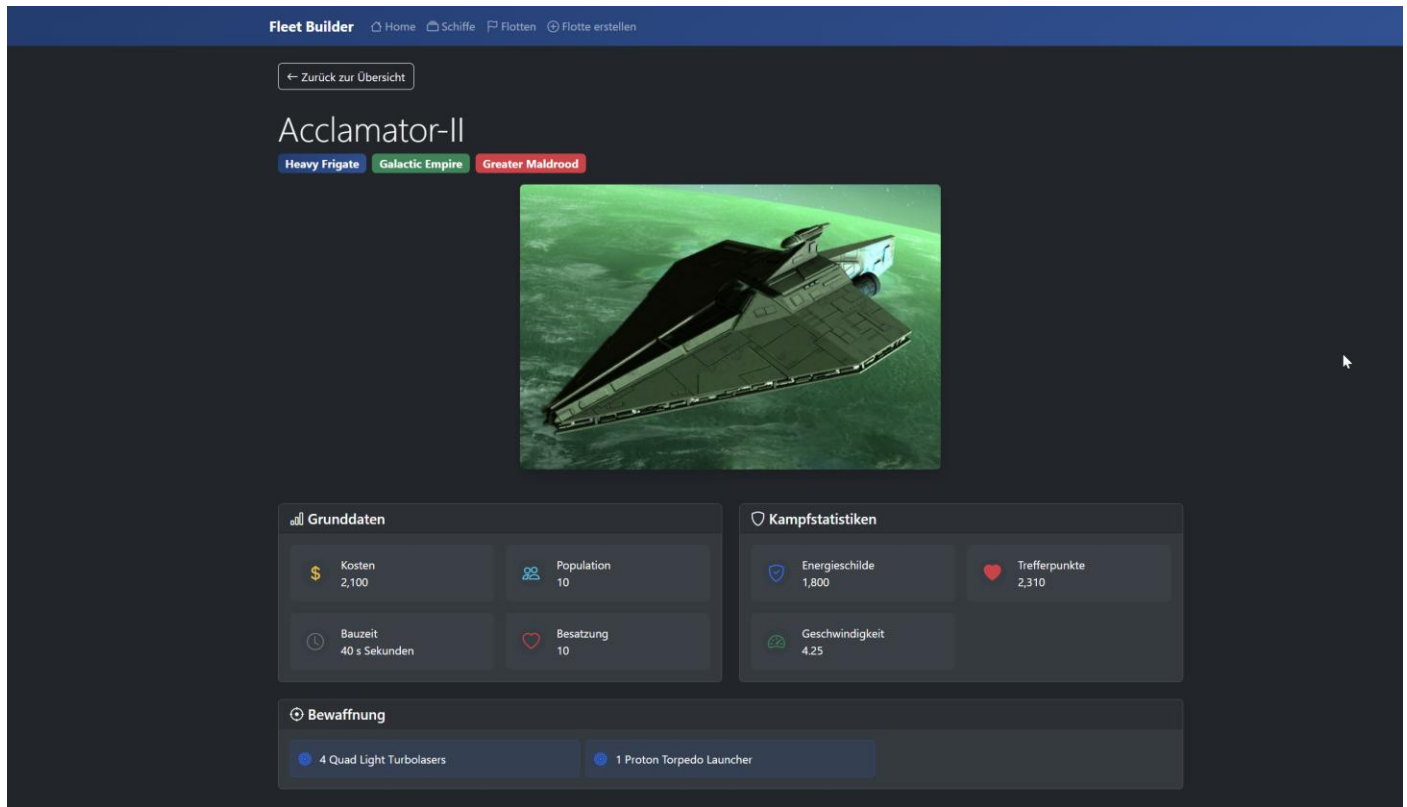
Auf dieser Seite sind sämtliche Schiffe ersichtlich. Durch die Filterfunktionen kann diese Ansicht eingeschränkt werden. Durch das anklicken eines Schiffstyps gelangt man zu den Detailinformationen.

Dateien:

- routes/ships/+page.svelte - Schiffe-Übersicht mit Filter und Suche
- routes/ships/+page.server.js - Lädt Schiffsdaten, Fraktionen und Funktionen
- routes/+layout.svelte - Navigation und Layout
- lib/components/ShipCard.svelte - Einzelne Schiffskarte
- lib/db.js - Datenbankfunktionen (getShips, getFactions, getShipTypes)
- lib/db-fallback.js - Fallback-Daten

### 3.3. Schiffe Detailansicht

Route: /ships/[id]



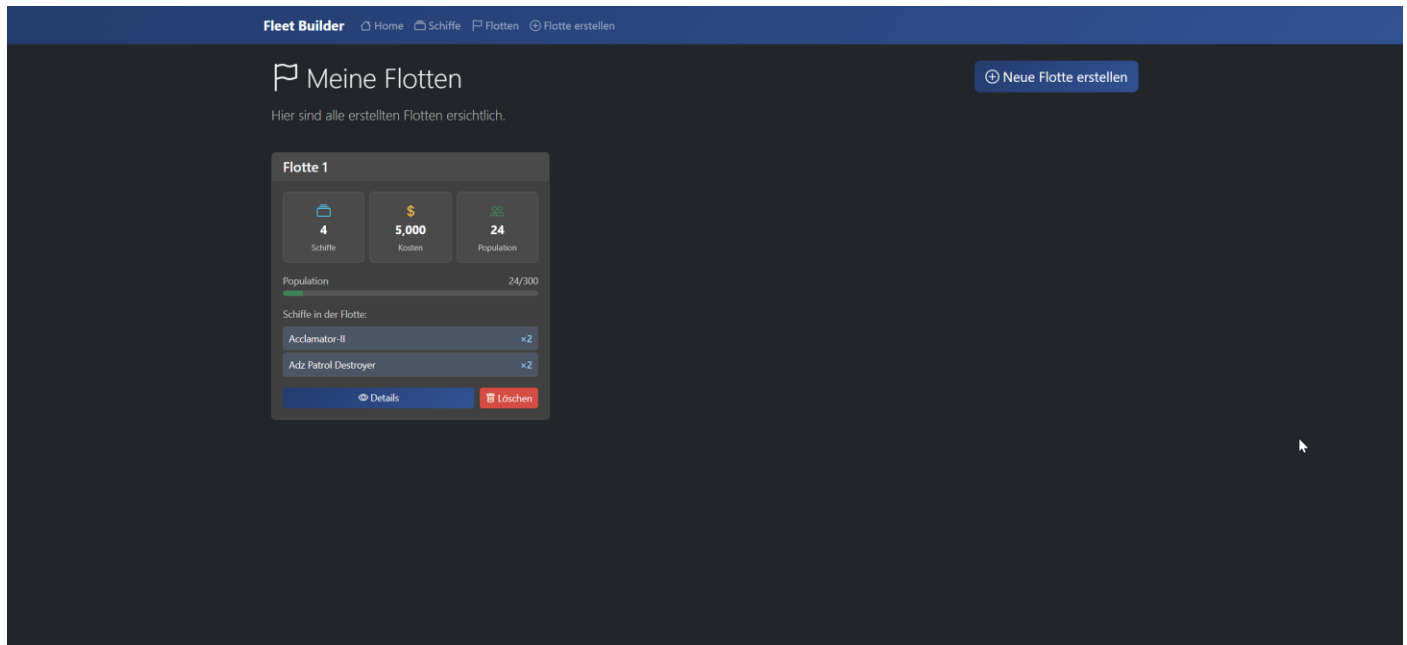
Auf dieser Seite sind mehr Informationen über die Schiffe verfügbar.

Dateien:

- routes/ships/[id]/+page.svelte - Schiff-Detailansicht
- routes/ships/[id]/+page.server.js - Lädt einzelnes Schiff nach ID
- routes/+layout.svelte - Navigation und Layout
- lib/db.js - Datenbankfunktion (getShipById)
- lib/db-fallback.js - Fallback-Daten

### 3.4. Flotten

Route: /fleets



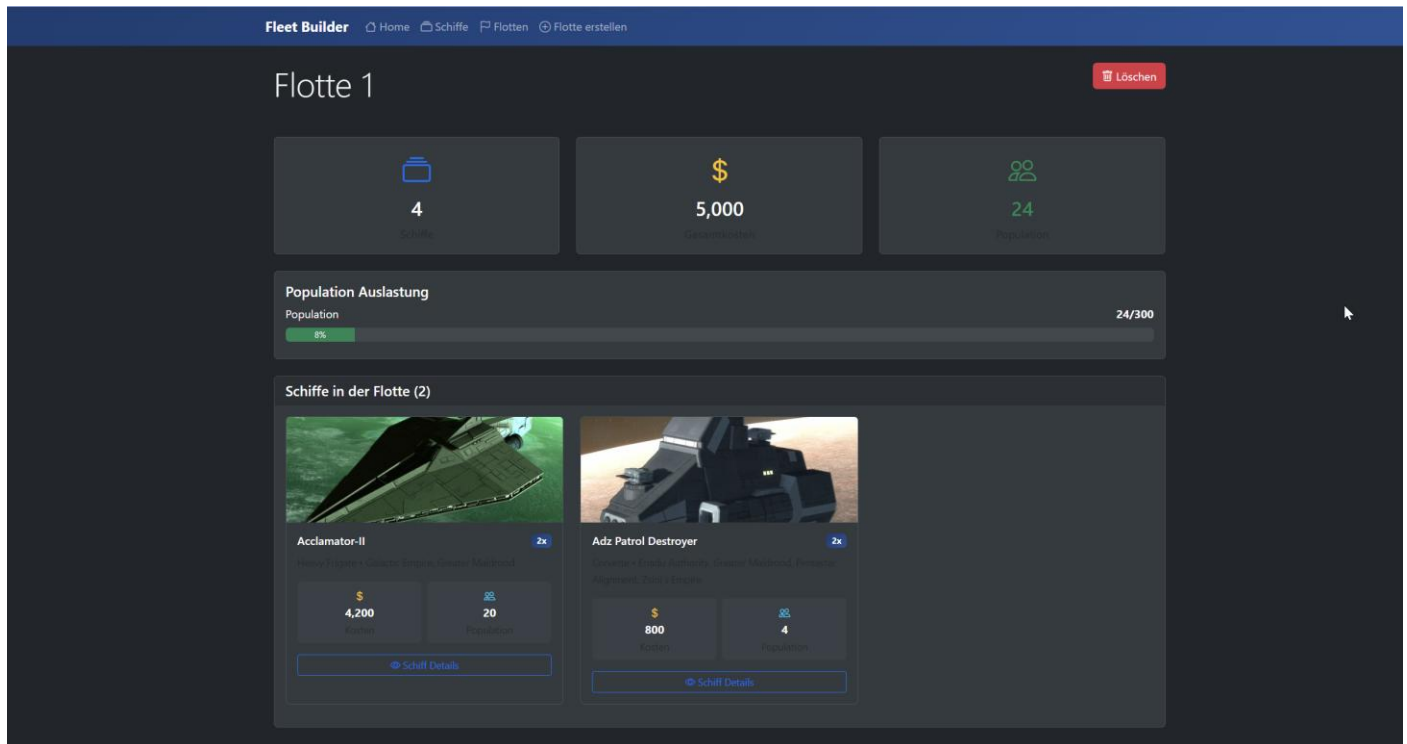
Auf dieser Seite sind alle gespeicherten Flotten ersichtlich. Man sieht auf den ersten Blick wie gross und teuer die Flotten sind. Hier kann man auch Flotten löschen. Durch das anklicken des Button Details kommt man zur grösseren und detaillierteren Ansicht dieser Flotte.

Dateien:

- routes/fleets/+page.svelte - Flotten-Übersicht mit Kacheln
- routes/fleets/+page.server.js - Lädt alle Flotten
- routes/+layout.svelte - Navigation und Layout
- lib/db.js - Datenbankfunktion (getFleets)
- lib/db-fallback.js - Fallback-Daten
- routes/api/fleets/[id]/+server.js - DELETE API für Löschen-Funktion

### 3.5. Flotten Detailansicht

Route: /fleets[id]



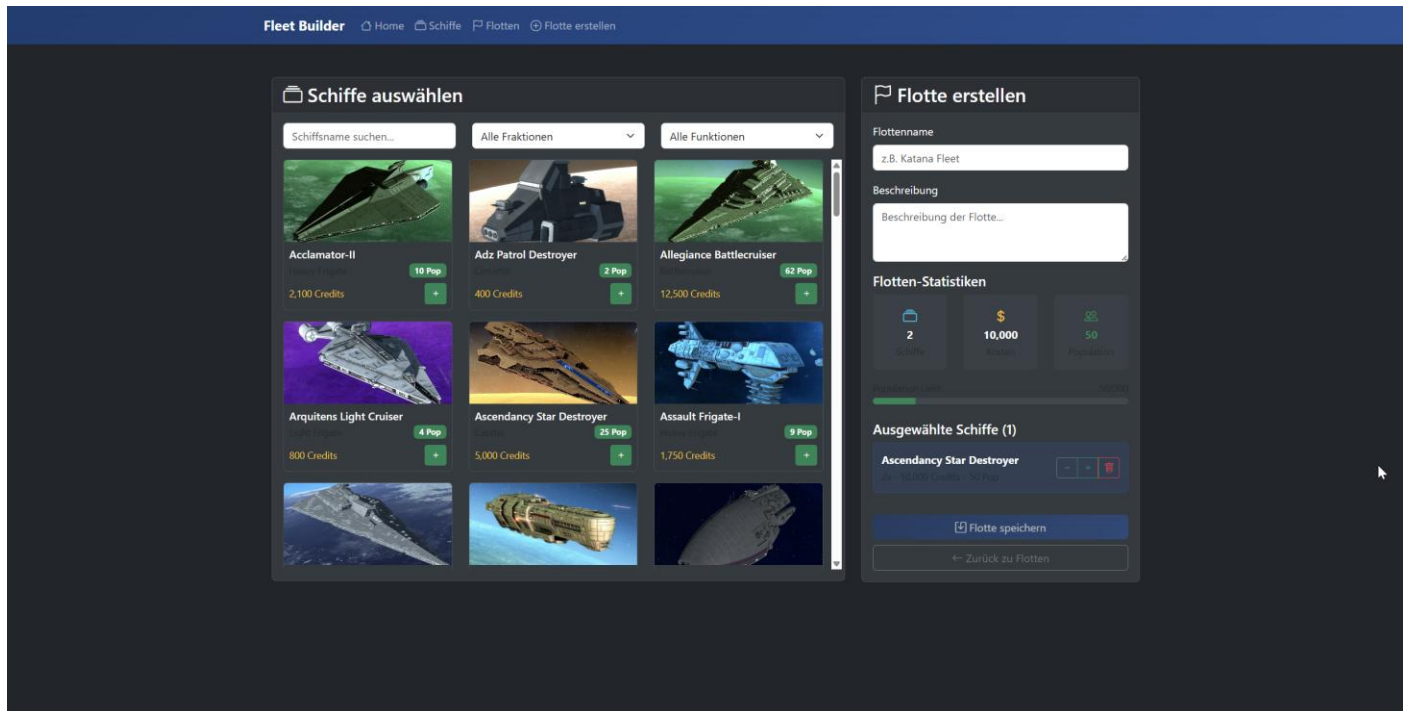
Auf dieser Seite sind alle Informationen dieser Flotte ersichtlich. Es kann auch von hier gelöscht werden. Ebenfalls können die Schiffe wie durch die Übersicht Schiffe genauer untersucht werden.

Dateien:

- routes/fleets/[id]/+page.svelte - Flotten-Detailansicht
- routes/fleets/[id]/+page.server.js - Lädt Flotte und alle Schiffe
- routes/+layout.svelte - Navigation und Layout
- lib/db.js - Datenbankfunktionen (getFleetById, getShips)
- lib/db-fallback.js - Fallback-Daten
- routes/api/fleets/[id]/+server.js - DELETE API für Löschen-Funktion

### 3.6. Flotte erstellen

Route: /builder



Auf dieser Seite können neue Flotten erfasst werden. Die Schiffe können nach belieben gefiltert werden, um das Suchen zu vereinfachen. Ebenfalls sieht man das Total der Schiffe, der Credits und der Population direkt. Hier kommt auch das Populations Limit ins Spiel. Man kann nur Schiffe bis maximal 300 Population anwählen.

Dateien:

- routes/builder/+page.svelte - Fleet Builder Interface
- routes/builder/+page.server.js - Lädt Schiffe, Fraktionen und Funktionen
- routes/+layout.svelte - Navigation und Layout
- lib/db.js - Datenbankfunktionen (getShips, getFactions, getShipTypes)
- lib/db-fallback.js - Fallback-Daten
- routes/api/fleets/+server.js - POST API zum Speichern von Flotten

## 4. Erweiterungen

### 4.1. Builder

Hat diverse ergänzende Funktionen wie Addition.

Datei(en):

- routes/builder/+page.svelte - Fleet Builder Interface
- routes/builder/+page.server.js - Lädt Schiffe, Fraktionen und Funktionen
- routes/+layout.svelte - Navigation und Layout
- lib/db.js - Datenbankfunktionen (getShips, getFactions, getShipTypes)
- lib/db-fallback.js - Fallback-Daten
- routes/api/fleets/+server.js - POST API zum Speichern von Flotten