

Prototyping Projektdokumentation

Name: Nikola Bogdanovic

E-Mail: bogdanik@students.zhaw.ch

URL der deployten Anwendung: <https://find-my-ride.netlify.app>

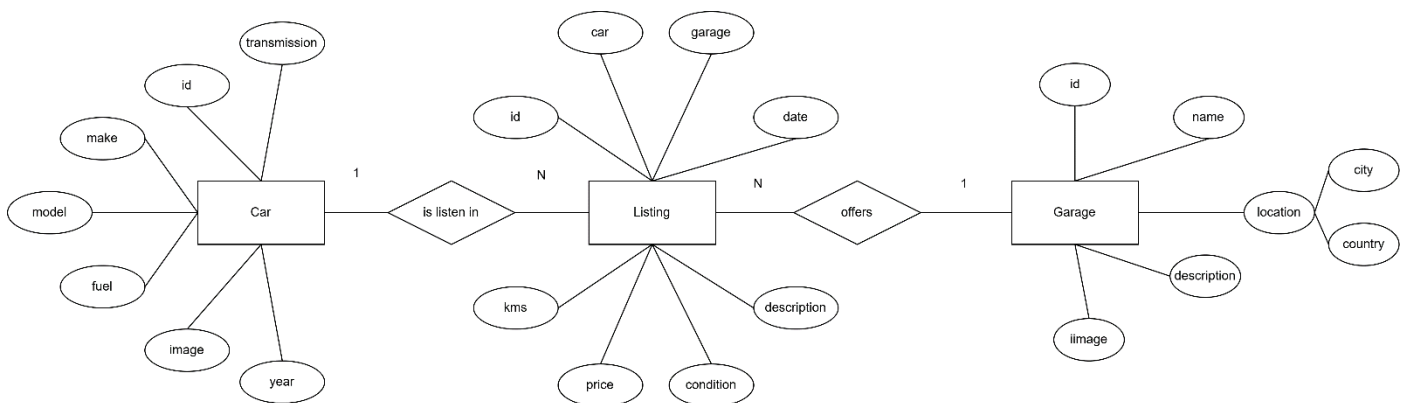
1. Einleitung

Find My Ride ist eine webbasierte Plattform zur Vermittlung und Verwaltung von Autoanzeigen. Die Anwendung ermöglicht es Nutzer:innen, vorhandene Garagen anzusehen, in diesen Garagen gelistete Fahrzeuge (Listings) zu durchsuchen und selbst neue Fahrzeuganzeigen einzustellen. Kernfunktionen sind:

- Anzeige aller gelisteten Fahrzeuge mit Detailinformationen
- Verwaltung und Anzeige von Garagen, in denen Fahrzeuge angeboten werden
- Eingabeformular zum Erstellen neuer Fahrzeuganzeigen
- Übersichtliche Benutzeroberfläche mit ansprechendem Design und Mobiloptimierung

2. Datenmodell

Die Datenbank basiert auf einer MongoDB mit drei zentralen Entitäten: Car, Garage und Listing. Ein Listing verknüpft jeweils genau einen Datensatz aus der Sammlung Car mit genau einem Datensatz aus der Sammlung Garage.



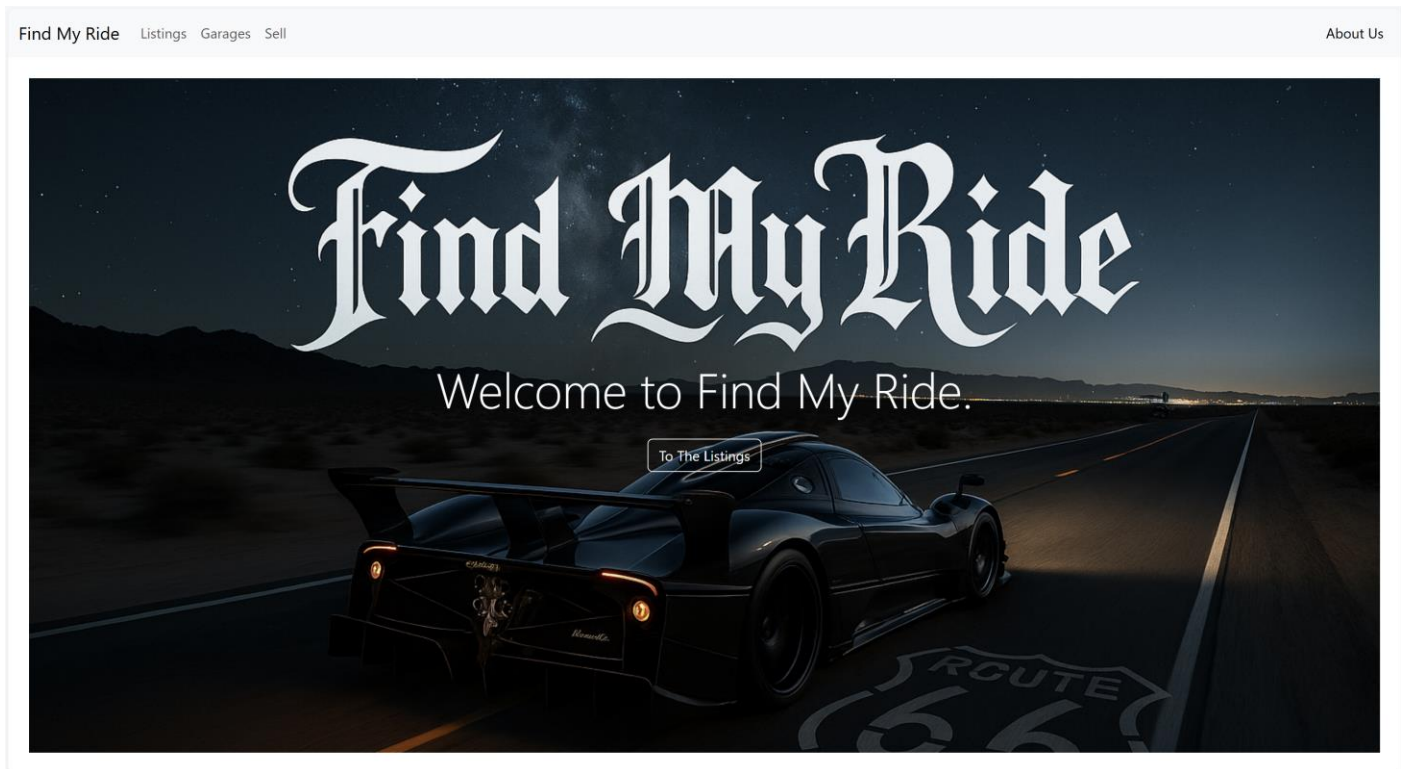
3. Erläuterungen:

- Car: Stellt einzelne Fahrzeugmodelle dar. Jedes Car-Dokument enthält u. a. Marke, Modell, Baujahr, Kraftstoffart, Getriebe und einen Bildpfad.
- Garage: Repräsentiert Werkstätten oder Verkaufsstellen, in denen Fahrzeuge angeboten werden können. Jedes Garage-Dokument enthält mindestens einen Namen, eine Ortsangabe, ein Bild und eine Beschreibung.
- Listing: Eine Fahrzeuganzeige, die jeweils exakt einen Car- und einen Garage-Datensatz referenziert. Die Referenzen werden beim Speichern in Form von eingebetteten (embedded) Objekten gehalten, sodass alle nötigen Kenndaten (z. B. Bildpfad, Marke/Modell, Name der Garage) direkt verfügbar sind. Ein Listing verfügt zusätzlich über Kilometerstand (kms), Preis, Fahrzeugzustand, eine Freitextbeschreibung und das Erstellungsdatum.

4. Beschreibung der Anwendung

4.1. Startseite

Route: /



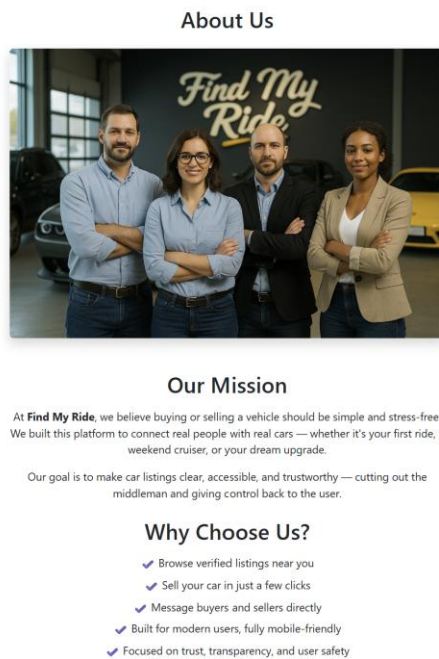
Die Startseite präsentiert einen grossflächigen Hero-Bereich mit dem Hintergrundbild „titelbild5.png“ und begrüsst die Nutzer:innen mit dem Schriftzug „Welcome to Find My Ride.“, darunter befindet sich ein Button mit der Aufschrift „To The Listings“, über den man direkt zur Übersichtsseite aller Fahrzeuganzeigen (/listings) gelangt.

Dateien:

- `src/routes/+page.svelte`

4.2. Über uns (About)

Route: /about



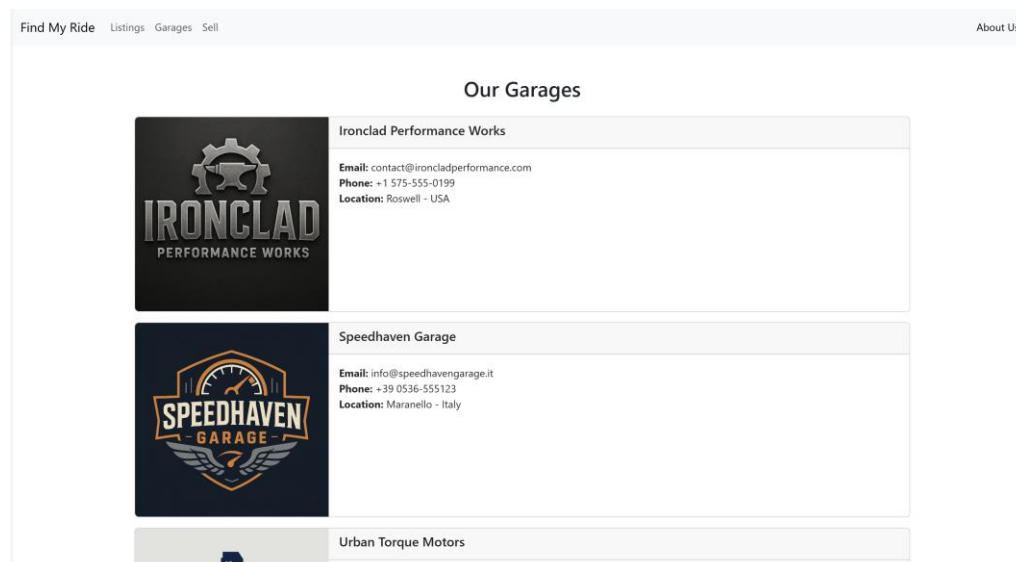
Auf der „About Us“-Seite erscheint zunächst die Überschrift „About Us“, gefolgt von einem Bild („aboutus.png“) und einem erläuternden Fliesstext, der die Mission von Find My Ride beschreibt: Fahrzeuge direkt zu vermitteln, ohne Zwischenhändler, und dabei Vertrauen sowie Benutzerfreundlichkeit in den Mittelpunkt zu stellen. Darunter werden mehrere Vorteile der Plattform aufgeführt, beispielsweise „Browse verified listings near you“ oder „Sell your car in just a few clicks“, die in Stichpunkten dargestellt sind.

Dateien:

- `src/routes/about/+page.svelte`

4.3. Garagenübersicht

Route: /garages



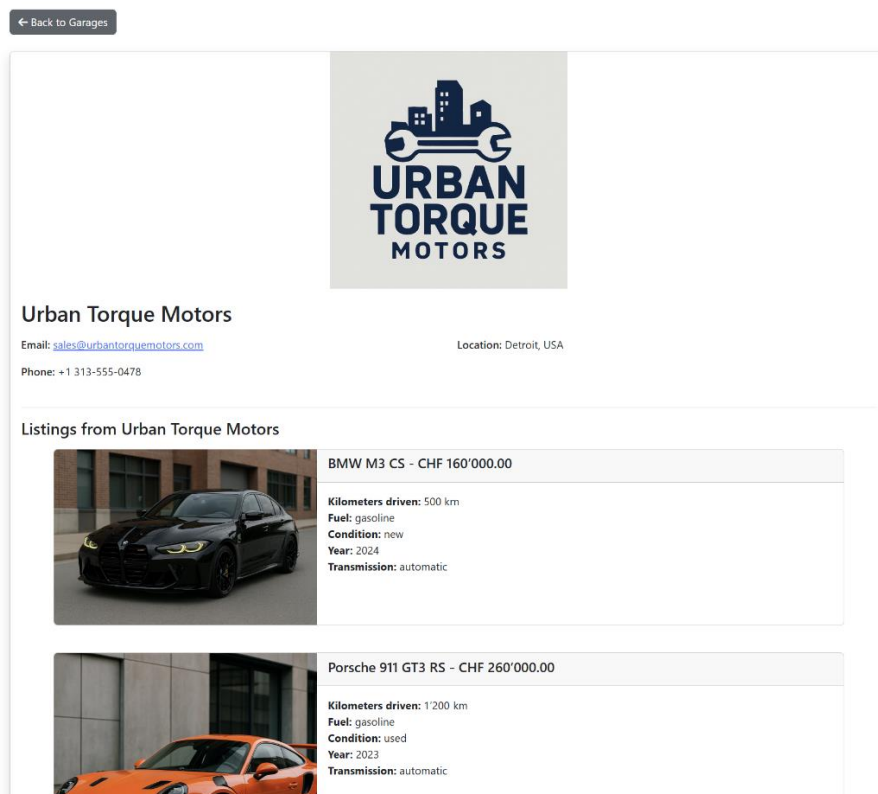
Auf der Garagenübersichtsseite werden in der Server-Load-Funktion von `+page.server.js` alle Garagen über den Aufruf von `getAllGarages()` aus der MongoDB geladen. Anschliessend wird für jede Garage eine `GarageCard`-Komponente gerendert, die den Namen der Garage, ihr Bild und ihren Standort anzeigt. Nutzer:innen haben die Möglichkeit, auf eine einzelne Garage zu klicken, um zur Detailseite `/garages/[id]` zu gelangen.

Dateien:

- `src/routes/garages/+page.server.js`
- `src/routes/garages/+page.svelte`
- `src/lib/components/GarageCard.svelte`

4.4. Garagendetail

Route: `/garages/[id]`






Über die Funktion `getGarage(id)` werden die Details der ausgewählten Garage (Name, Standort, Beschreibung und Bild) geladen, parallel dazu holt `getAllListingsOfGarage(id)` alle zugehörigen Fahrzeuganzeigen. Die Garage wird oben in einer Karte mit ihrem Bild (CSS-Klasse `.garage-banner`) dargestellt, darunter werden alle Listings dieser Garage mithilfe der `ListingCard`-Komponente aufgelistet, die jeweils Bild, Marke/Modell, Preis und Kilometerstand anzeigt. Ein „Back to Garages“-Button am Seitenende führt zurück zur Übersichtsseite `/garages`.

Dateien:

- `src/routes/garages/[id]/+page.server.js`
- `src/routes/garages/[id]/+page.svelte`
- `src/lib/components/ListingCard.svelte` (für Anzeigen innerhalb der Garage)
- `src/lib/util.js` (Formatierung von Datum, Preis, Abstand)

4.5. Listingsübersicht

Route: /listings

Find My Ride	Listings	Garages	Sell	About Us
<h3>Available Listings</h3>				
				
Ferrari Enzo - CHF 2'200'000.00				
Kilometers driven: 4'500 km Fuel: gasoline Condition: used Year: 2002 Transmission: automatic				
				
Pagani Zonda R - CHF 4'500'000.00				
Kilometers driven: 200 km Fuel: gasoline Condition: used Year: 2009 Transmission: manual				
				
Lamborghini Countach 25th Anniversary - CHF 900'000.00				
Kilometers driven: 6'000 km Fuel: gasoline Condition: used				

Über die Server-Load-Funktion wird mit `getAllListings()` eine Liste aller Anzeigen aus der Datenbank abgefragt. Anschliessend wird für jede Anzeige eine `ListingCard`-Komponente gerendert, die das Vorschaubild des Fahrzeugs (`listing.car.image`), Marke und Modell, Preis, Kilometerstand, Baujahr, Kraftstoffart, Getriebe und Zustand übersichtlich darstellt. Nutzer:innen können auf eine dieser `ListingCards` klicken, um zur Detailseite `/listings/[id]` zu gelangen.

Dateien:

- `src/routes/listings/+page.server.js`
- `src/routes/listings/+page.svelte`
- `src/lib/components/ListingCard.svelte`
- `src/lib/util.js` (Formatierungsfunktionen)

4.6. Listingdetail

Route: /listings/[id]

← Back to Listings



TR-3B Triangle Stealth Craft – CHF 999'999'999'999'999'999.00

Delete Listing

Kilometers: 0 km

Condition: New

Fuel Type: Element 115

Year: 9999

Transmission: Antigravity

Posted on: 1. Juni 2025

Description

Found in the desert near Area 51 - still looking for parking!



Listed by Garage

Name: Ironclad Performance Works

Email: contact@ironcladperformance.com

Phone: +1 575-555-0199

Location: Roswell, USA

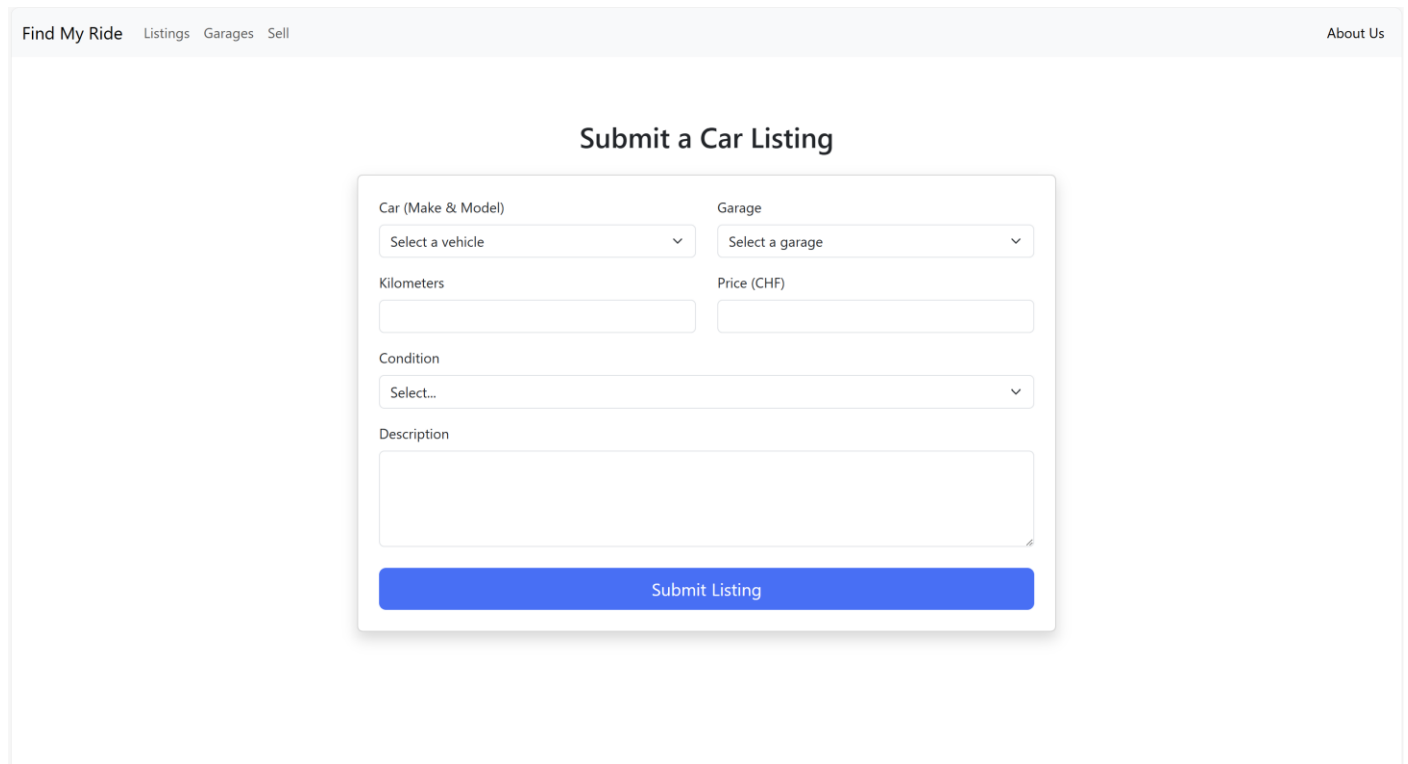
Mit der Funktion `getListing(id)` werden alle Daten des ausgewählten Listings aus der Datenbank geladen, wobei sowohl das eingebettete Car-Objekt als auch das zugehörige Garage-Objekt bereitgestellt werden. Auf der Detailseite wird zunächst eine grosse Fahrzeugabbildung angezeigt, gefolgt von den wichtigsten Kenndaten wie Marke/Modell, Baujahr, Kilometerstand, Preis, Kraftstoffart, Getriebe, Zustand, Beschreibung und dem Datum der Erstellung. Unterhalb dieser Fahrzeuginformationen befindet sich ein Abschnitt „Garage“, der den Namen, den Standort und ein Bild der Garage zeigt, in der das Fahrzeug gelistet ist – ein Klick auf diesen Abschnitt führt zur entsprechenden Garagendetailseite (`/garages/[garage._id]`). Ausserdem gibt es ein Formular mit einem „Delete Listing“-Button; nach Absenden wird `deleteListing(listing_id)` ausgeführt und man wird anschliessend zurück zur Listings-Übersicht (`/listings`) weitergeleitet. Als kleines Easter Egg wird beim Aufruf einer bestimmten `car._id` (z. B. „683cbae7946b73a8bdc377a“) automatisch das Audio-File `/xfilestheme.mp3` abgespielt.

Dateien:

- `src/routes/listings/[id]/+page.server.js`
- `src/routes/listings/[id]/+page.svelte`
- `src/lib/util.js`

4.7. Anzeige erstellen (Sell Page)

Route: /sell



The screenshot shows a web application interface for submitting a car listing. At the top, there is a navigation bar with links: 'Find My Ride', 'Listings', 'Garages', 'Sell', and 'About Us'. The main heading is 'Submit a Car Listing'. Below this, there is a form with the following fields:

- Car (Make & Model):** A dropdown menu with the placeholder text 'Select a vehicle'.
- Garage:** A dropdown menu with the placeholder text 'Select a garage'.
- Kilometers:** A text input field.
- Price (CHF):** A text input field.
- Condition:** A dropdown menu with the placeholder text 'Select...'.
- Description:** A large text area for a detailed description.

At the bottom of the form is a blue button labeled 'Submit Listing'.

Die Seite lädt mithilfe von `getAllCars()` und `getAllGarages()` alle verfügbaren Autos und Garagen, um sie anschliessend in Dropdown-Menüs anzuzeigen.

Formularfelder:

- Auswahlfeld „Car (Make & Model)“ (Name/Modell der Fahrzeuge)
- Auswahlfeld „Garage“ (Name der Garage)
- Eingabefeld „Kilometers“ (Kms)
- Eingabefeld „Price“ (Preis in CHF)
- Auswahlfeld „Condition“ („New“, „Used“, „Broken“)
- Textarea „Description“ (Freitextbeschreibung)

Beim Absenden (`method="POST"`) wird ein neues Listing-Objekt erstellt, das zusätzlich ein `date`-Feld mit dem aktuellen Datum enthält. Anschliessend ruft die `actions.default`-Funktion `createListing(listing)` auf und leitet per Redirect zurück zur /listings-Übersicht.

Dateien:

- `src/routes/sell/+page.server.js`
- `src/routes/sell/+page.svelte`
- `src/lib/db.js` (enthält `createListing`, `getAllCars`, `getAllGarages`)

5. Erweiterungen

Im Vergleich zu den in den Grundanforderungen geforderten Basisfunktionen wurden folgende zusätzliche Features implementiert:

5.1. „About Us“-Seite

Unter `/about` wurde eine ergänzende statische Informationsseite eingerichtet, die über die Basisanforderungen hinausgeht. Sie enthält neben einem erklärenden Fliesstext auch ein Bild und eine Auflistung von Vorteilen, um die Mission der Plattform anschaulich zu vermitteln. Die gesamte Logik und das Layout sind in `src/routes/about/+page.svelte` implementiert.

5.2. Easter Egg Audio-Integration

In der Detailansicht eines Listings (`/listings/[id]`) wird bei einem bestimmten Fahrzeug (identifiziert über die statische `car_id == "683cbae7946b73a8bdc377a"`) automatisch das Theme der X-Files (`/xfilestheme.mp3`) abgespielt. Die entsprechende Logik ist in `src/routes/listings/[id]/+page.svelte` implementiert.

5.3. Löschfunktion für Listings

Auf der Detailseite eines Listings (`/listings/[id]`) besteht die Möglichkeit, eine bestehende Anzeige direkt zu löschen. Dazu wird eine Formular-Action genutzt, die `deleteListing(listing_id)` in der Datei `src/routes/listings/[id]/+page.server.js` aufruft.

5.4. Formatierungsfunktionen und Utility-Bibliothek

Gemeinsame Hilfsfunktionen in `src/lib/util.js` für:

- `formatPrice`: Formatierung in Schweizer Franken (CHF).
- `formatDistance`: Formatierung von Kilometern (z. B. „15 km“).
- `formatDate`: Datumsausgabe im Format „DD. Monat YYYY“.
- `capitalizeString`: Grossschreibung des ersten Buchstabens bei Strings.

Diese Utility-Funktionen verbessern die Konsistenz der Anzeige und wurden in mehreren Komponenten wiederverwendet.