

Prototyping Projektdokumentation

Name: Yasemin Anik

E-Mail: anikyas1@students.zhaw.ch

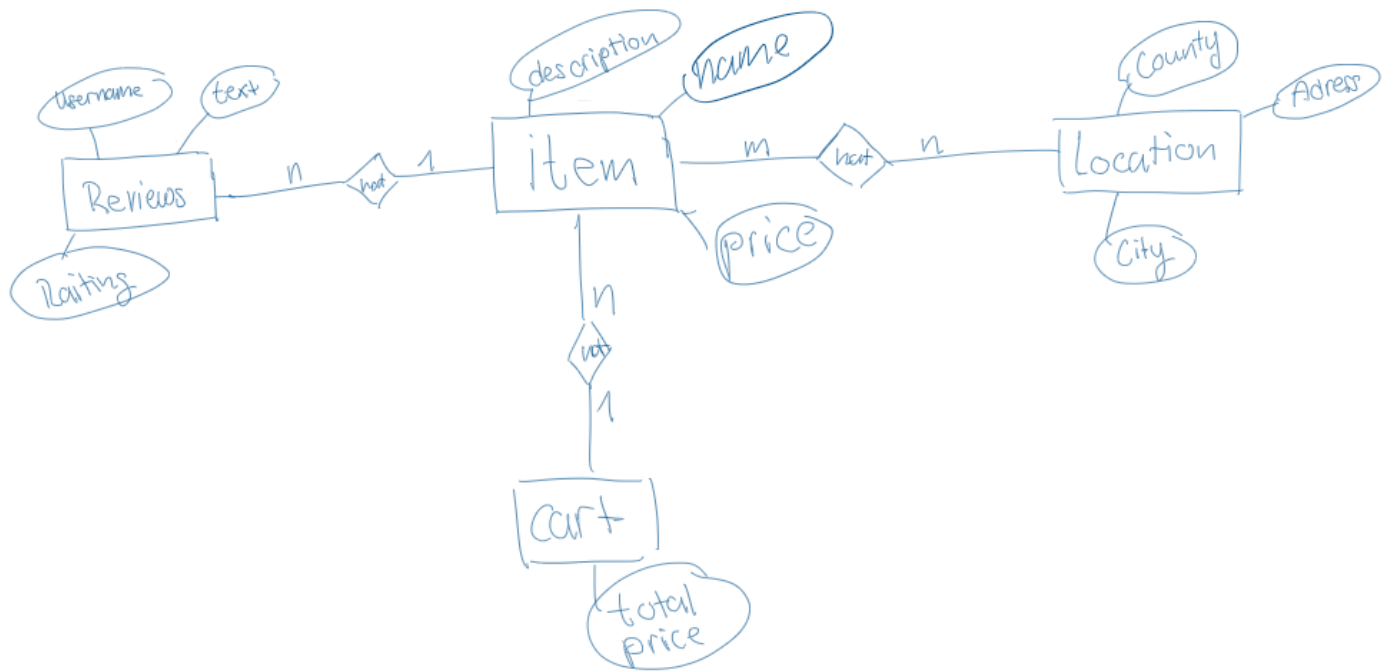
URL der deployten Anwendung: <https://mywebshopln2.netlify.app/>

Einleitung

Meine Anwendung heisst „Shopping Today“ und ist ein einfacher Webshop, in dem man viele Produkte finden und kaufen kann. Natürlich kann man hier nichts kaufen, da die Artikel nicht existieren.

Die Grundfunktionen ähneln denen, die man typischerweise in einem Online-Shop erwarten würde. Die erstellte Ansicht ist die eines internen Administrators, d.h. man kann neue Produkte hinzufügen und bestehende löschen. Man kann Rezensionen hinzufügen, Produktdetailseiten anzeigen, Produkte in den Warenkorb legen und auch wieder entfernen. Ausserdem kann man sich die Standorte anzeigen lassen. Die Navigation ist neben der Navigationsleiste oben auch über die Suchleiste möglich.

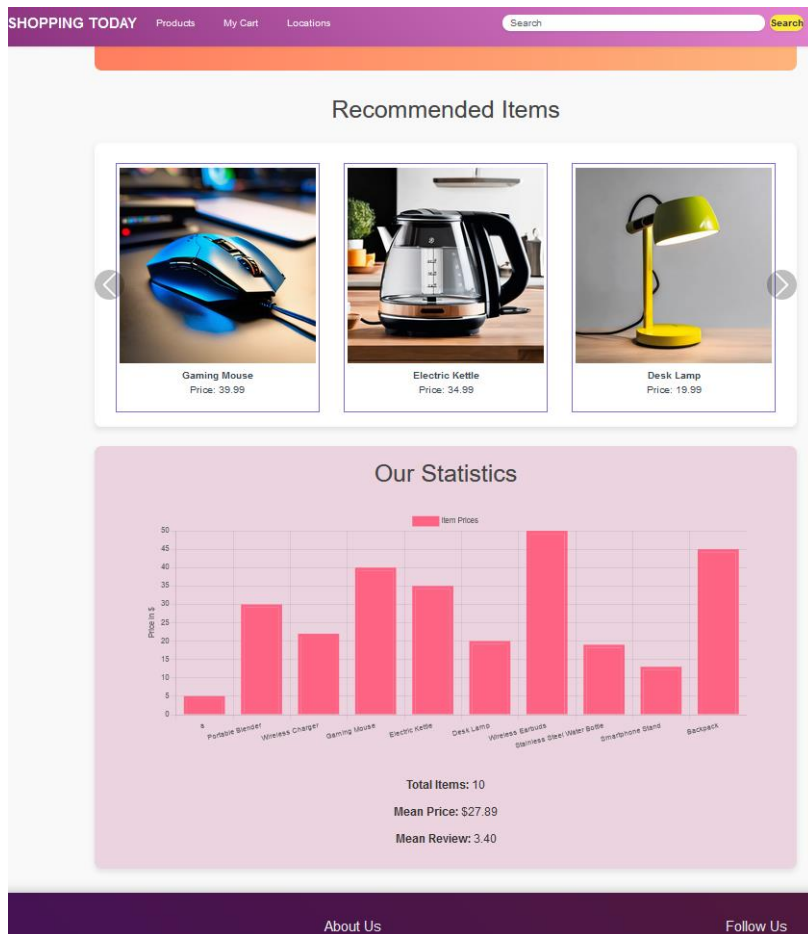
Datenmodell



Beschreibung der Anwendung

Startseite

Route: /



Oben und unten befinden sich die Navigationsleiste bzw. die Fusszeile, die in der `+layout.svelte` definiert wurden und entsprechend auf jeder der folgenden Seiten erscheinen sollten. In der Navigationsleiste befindet sich auch eine Suchleiste, die die Navigation erleichtern soll und über die Elemente in der Navigationsleiste gesucht werden kann.

Auf dieser Seite wird eine zufällige Liste der vorgeschlagenen Produkte in einem Karussell angezeigt. Durch Auswahl des Pfeils wird eine neue Reihe von Produkten angezeigt. Die einzelnen Produkte werden mit Hilfe der Komponente `Items.svelte` dargestellt, gemäss dieser Komponente führt das Anklicken eines beliebigen Teils des Produkts zu dessen Detailansicht, also zum Beispiel zu `«Items/6776a7a3eb3b84f6e804ee78»`. Zudem befindet sich noch eine Statistik inklusive Chart am unteren Ende der Seite.

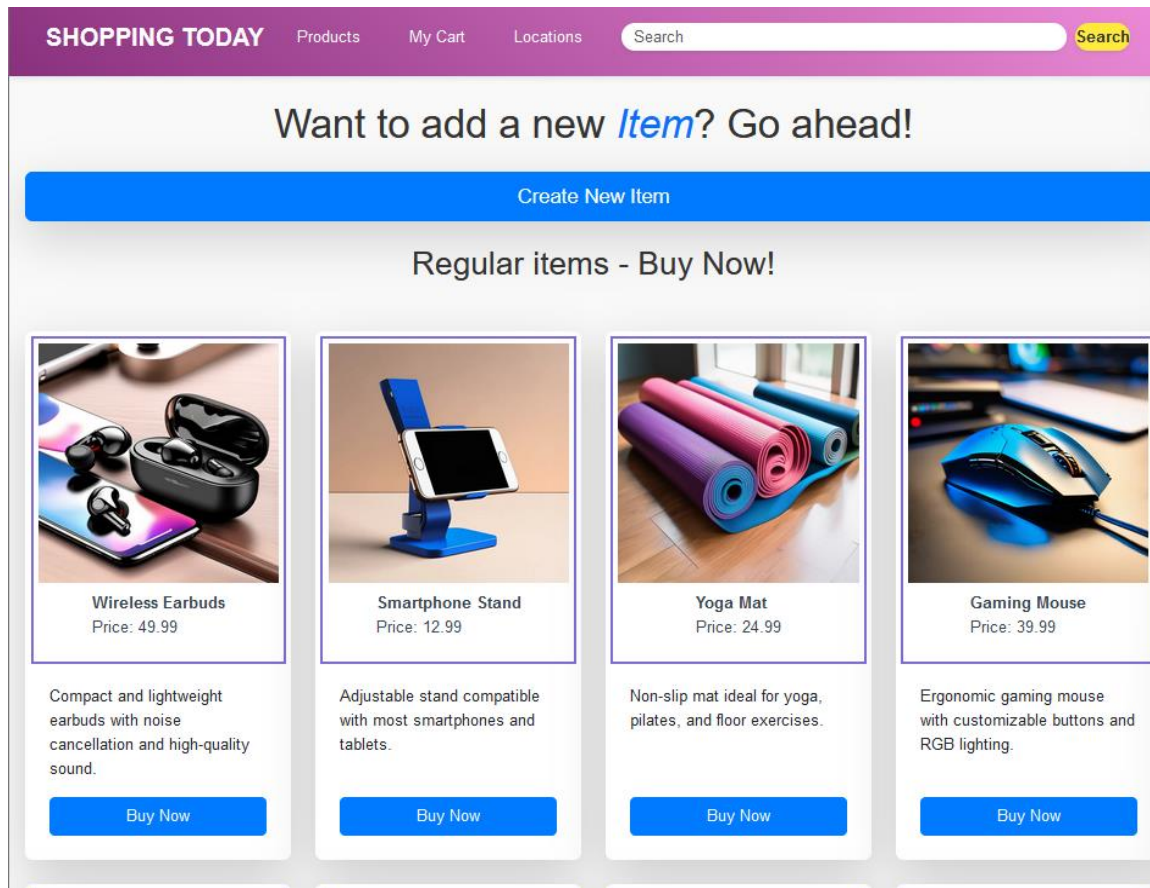
Dateien:

Lib/Components/items.svelte

Lib/Routen/+page.svelte

Lib/routen/+layout.svelte

Lib/routes/+page.server.js



Auf dieser Seite gibt es oben, wenn auch etwas ungewöhnlich, einen Button "Create New Item", der den Nutzer zu einem Formular unter «/Items/itemcreate» führt, welches ihm erlaubt neue Produkte hinzuzufügen. Danach kommen wir zu einer Liste aller Produkte, die laut MongoDB in dieser Collection existieren. Der blau umrahmte Teil ist auch hier wieder die Items.svelte Komponente, dementsprechend führt auch hier ein Klick auf die Komponente zur entsprechenden Detailansicht. Alternativ kann man auch auf den «Buy Now» Button klicken, um ebenfalls zur Detailansicht zu gelangen.

Dateien:

Lib/Components/items.svelte

Lib/Routen/ Items/+page.svelte

Lib/routes/ Items/+page.server.js

Add an Item

Route: /Items/itemcreate

The screenshot shows a web application interface for adding a new item. At the top is a purple navigation bar with the text 'SHOPPING TODAY' and links for 'Products', 'My Cart', and 'Locations'. A search bar with a 'Search' button is also present. Below the navigation bar, there is a 'Back' link. The main heading is 'Add an Item'. The form consists of several fields: 'Name' (a text input), 'Price' (a text input with a currency symbol icon), 'Description' (a large text area), 'Warnings' (a list of checkboxes for 'Flammable', 'Fragile', 'Heavy', and 'Perishable'), and 'Image File Path' (a text input with a placeholder example 'e.g., /images/item.jpg'). A blue 'Add Item' button is at the bottom of the form. A small note below the warnings says 'Hold down the Ctrl key to select multiple warnings.'

Hier sieht man das oben erwähnte Formular, man kann die Felder Name, Preis, Beschreibung und Bild Url definieren. Name und Preis sind Pflichtfelder. Wenn die Erstellung erfolgreich war, wird eine Meldung angezeigt, die die Erstellung bestätigt. Mit dem «Back» Button oben kommt man zurück auf die «/Items» Seite.

Dateien:


Lib/Routen/ Items /itemcreate/+page.svelte

Lib/routes/ Items /itemcreate/+page.server.js

Detailansicht

Route: /Items/[_id]

Bsp. Items/6776a7a3eb3b84f6e804ee7b



Gaming Mouse
Price: \$39.99
Ergonomic gaming mouse with customizable buttons and RGB lighting.
★ Average Rating: 4.5 / 5
Remove from Cart

Available for pickup in..

Dortmund
Address: Suite 91
County: Germany

Winterthur
Address: Apt 1918
County: Switzerland

Edinburgh
Address: Room 477
County: United Kingdom

Reviews

Write Review

GamerPro (Rating: 5/5): Responsive and comfortable for long gaming sessions.

RGBFan (Rating: 4/5): Love the lights, but the software could be better.

Delete Item

Dies ist die oben erwähnte Detailansicht. Sie besteht aus vier Teilen.

Zuerst eine Ansicht des Produkts mit Bild, Name, Preis, Beschreibung, durchschnittlicher Bewertung und einem Button, der entweder «Add to Cart» oder, falls sich das Produkt bereits im Warenkorb befindet, «Remove from Cart» anzeigt.

Als nächstes folgt ein Abschnitt, der alle Standorte auflistet, an denen das Produkt direkt im Geschäft gekauft werden kann. Für die Auflistung wird auf die Komponente `locations.svelte` zugegriffen.

Als nächstes werden alle vorhandenen Reviews für das Produkt angezeigt. Eine Bewertung besteht aus dem Benutzernamen, einer Bewertung von 1-5 und einem Text. Die obere Durchschnittsbewertung passt sich bei jeder neuen Bewertung automatisch an. Wer selbst eine Bewertung schreiben möchte, kann dies über den Button «Bewertung schreiben» tun. Dieser führt den Benutzer auf die Seite «/create», auf der er ein Formular ausfüllen kann.

Zuletzt gibt es noch den «Delete Item» Button, der, wie der Name schon sagt, das Produkt löscht und zur «/» Startseite zurückführt.

Dateien:

Lib/Components/locations.svelte

Lib/Routen/Items/[_id]/+page.svelte

Lib/Routen/Items/[_id]/+page.server.js

Reviews

Route: /Items/[_id]/create

Bsp. Items/6776a7a3eb3b84f6e804ee7b/create

Write a Review

Username:

Anonymous

Rating:

Review:

Submit Review

Submit Review

YIPPY! Review has been successfully submitted!

Die folgende Seite dient der Erstellung von Reviews. Der Benutzername ist nach dem Standtyp auf Anonym gesetzt und die Bewertung auf 1. Das Textfeld «Review:» ist ein Pflichtfeld. Wenn die Abgabe des Reviews erfolgreich war, erscheint ein entsprechendes Banner. Das entsprechende Review wird dann in der Dateiansicht des Produktes angezeigt und die Durchschnittsbewertung entsprechend angepasst.


Dateien:

Lib/Routen/Items/[_id]/create/+page.svelte

Lib/Routen/Items/[_id]/create/+page.server.js

Your *Shopping Cart*

Items in your cart




Wireless Earbuds
Price: 49.99

Compact and lightweight earbuds with noise cancellation and high-quality sound.

\$49.99

Delete




Smartphone Stand
Price: 12.99

Adjustable stand compatible with most smartphones and tablets.

\$12.99

Delete




Gaming Mouse
Price: 39.99

Ergonomic gaming mouse with customizable buttons and RGB lighting.

\$39.99

Delete



Order Summary

Total Items: 4
Total Price: **\$122.96**

Proceed to Checkout

Keep Shopping

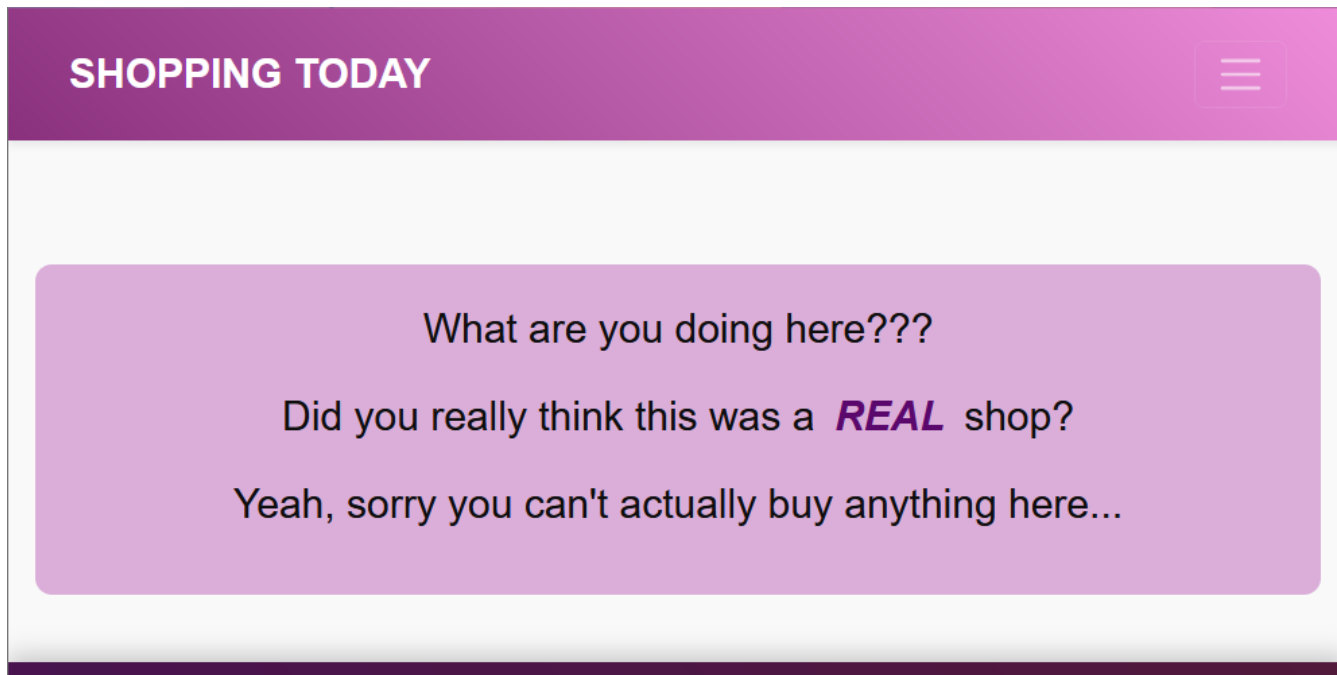
Auf dieser Seite befindet sich eine gefilterte Liste der Artikel, die nur die Artikel anzeigt, die in den Warenkorb gelegt wurden. Auch hier wurde die `Items.svelte` Komponente verwendet. Durch Anklicken des «Löschen»-Buttons können diese wieder entfernt werden. Auf der rechten Seite der Bestellübersicht gibt es zwei Kennzahlen. Zum einen wird angezeigt, wie viele Artikel sich insgesamt im Warenkorb befinden und zum anderen, wie viel diese Artikel zusammen kosten.

Der Button «Weiter einkaufen» bringt Sie zurück zur Produktseite. Der Button "Proceed to Checkout" führt natürlich nicht auf die Checkout-Seite, sondern auf `/FailSafe` eine Seite, die darüber informiert, dass dies kein echter Shop ist und man hier nichts kaufen kann.

Dateien:

Lib/Components/items.svelte

Lib/Routen/ Cart/+page.svelte



Dies ist die FailSafe Seite. Hier landet man nur wenn man zum Checkout will weil diese Funktion nicht existiert.


Dateien:

Lib/Routen/ FailSafe/+page.svelte

Locations

Route: /Locations

Our Locations



The map shows Europe with red location pins in Berlin, Bradford, and Cottbus. Below the map, the locations are listed in detail.

Berlin

Location	Address	County
Berlin	PO Box 77518	Germany
Berlin	12th Floor	Germany

Bradford

Location	Address	County
Bradford	9th Floor	United Kingdom
Bradford	Room 1171	United Kingdom
Bradford	Room 1863	United Kingdom

Cottbus

Location	Address	County
Cottbus	Room 94	Germany

Diese Seite listet alle Städte auf, die über mindestens ein physisches Geschäft verfügen. Wenn Sie auf die rote Markierung klicken, erscheint ein Popup das den Namen der Stadt anzeigt. Dort findet man eine nach Städtenamen sortierte Liste aller Standorte.

Dateien:

Lib/Components/locations.svelte

Lib/Routen/ Locations/+page.svelte

Erweiterungen

Startseite

Auf der Startseite «/» gibt es zwei Erweiterungen

Statistik inkl. Chart

In diesem Abschnitt gibt es eine kleine Statistik über die Produkte. In Form eines Säulendiagramms bekommt man hier die Preise für alle Produkte angezeigt. Darunter befinden sich 3 Berechnungen, eine für die Anzahl der Produkte, eine für den Durchschnittspreis der Produkte und eine für die durchschnittliche Bewertung der Produkte.

Dateien

Routes/+page.server.js

Routen/+page.svelte

Karussell

Das Karussell zeigt bei jedem Laden der Seite dank der function shuffle(array)

Immer genau 6 zufällige Produkte an.

Dateien

Routes/+page.server.js

Routes/+page.svelte

Function

function shuffle(array)

Reviews Formlar Reset

Das Felder beim Formular uner /Items/[_id]/create Bsp. Items/6776a7a3eb3b84f6e804ee7b/create werden nach Erfolgreicher Sendung der Bewertung wieder auf ihre Default werte gestellt.

Dateien:

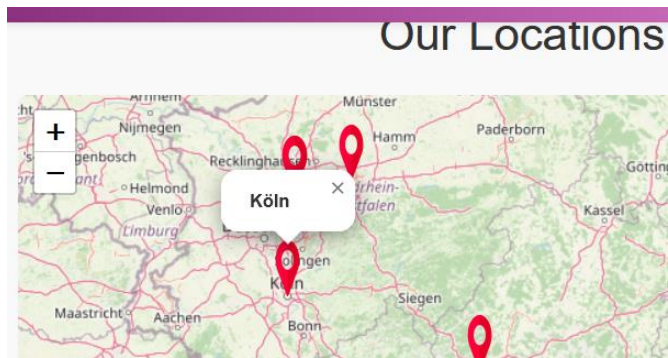
Lib/Routen/Items/[_id]/create/+page.svelte

Lib/Routen/Items/[_id]/create/+page.server.js

Location Seite

Auf der Seite /Locations befinden sich zwei Erweiterungen.

Karte



Die Koordinaten der Orte wurden mit Hilfe von Nominatim Api, einem kostenlosen Geocodierungsservice von Open StreetMap, basierend auf den Städtenamen ermittelt. Um die Karte zu visualisieren, ist zusätzlich die leaflet.js Bibliothek heruntergeladen worden. Dies ermöglicht das Erstellen einer interaktiven Karte, welche die Städte aus der MongoDB Collection mit Markern anzeigt und mit angebundenen popups beschriftet.

Dateien:

Routes/Locations/+page.svelte

Routes/Locations/+page.server.js

Funktionen:

```
async function geocodeCity(cityname)
```

```
async function initializeMap()
```

Sortierung Location

Für diese Seite habe ich zusätzlich die Komponente /Locations.svelte verwendet, diese mit Hilfe eines `{#each}` -Blocks alphabetisch sortiert und entsprechend der Sortierung eine Überschrift gegeben. So hat man immer einen Städtenamen als Überschrift und darunter alle Orte der jeweiligen Stadt.

Dateien:

Lib/Components/locations.svelte

Routes/Locations/+page.svelte

Routes/Locations/+page.server.js

Warenkorb

Die Seite /Cart hat 2 kleine Erweiterungen.

Filterung Warenkorb

Diese Seite ist eine gefilterte Liste, ähnlich wie die «Watchlist» aus dem Unterricht, nur dass diese auf einer separaten Seite angezeigt wird. die Liste kann zudem an zwei Stellen angepasst werden, bei den Produktdetails und direkt beim Warenkorb selbst.

Dataien

/routes/Cart/+page.svelte

Funktionen

async function removeFromCart(itemId)

Summierung Warenkorb

Es gibt zwei Berechnungen auf dieser Seite, die ebenfalls über einen Filter funktionieren und mir zum einen den Gesamtpreis des Warenkorbs zurückgeben und zum anderen die Gesamtanzahl der Artikel im Warenkorb zählen. Der Gesamtpreis wird hierbei anhand mit «.toFixed(2)» immer auf Kommastellen gerundet.

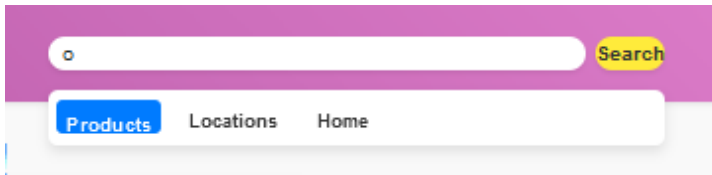
Dataien

/routes/Cart/+page.svelte

Layout seite

In Routes/+layout.svelte gibt es drei Erweiterungen.

Suchleiste



Die Navigationsleiste verfügt über eine funktionierende Suchleiste, macht Seitenvorschläge und leitet auf die richtige Seite weiter.

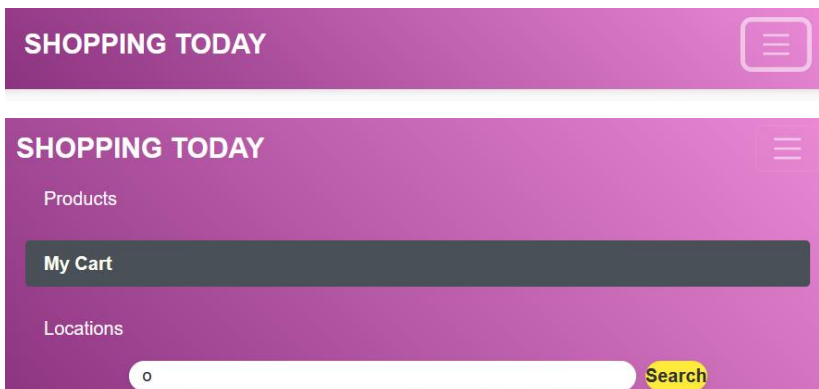
Dateien:

Routes/+layout.svelte

Funktionen:

```
function handleSearch(event)
function handleInput(event)
function getSuggestions(query)
function handleSuggestionClick(suggestion)
```

Navbar-Toggler



Beim Verkleinern der Seite werden die Elemente der Navigationsleiste hinter einem Toggler versteckt.

Dateien:

Routes/+layout.svelte

Laden von Bildern über Links

In der Fuszeile sind bilder welche nicht vom static/images sondern von einem url, bsp.

<https://cdn-icons-png.flaticon.com/512/733/733547.png>

Dateien:

Routes/+layout.svelte

Design

Die gesamte Anwendung hat ein einheitliches, übersichtliches und benutzerfreundliches Design, das mittels CSS und Bootstrap generiert wurde.

Im Wesentlichen sind folgende Dateien aussagekräftig, es gibt auf allen Seiten einen CSS-Style-Teil für seitenspezifische Darstellungsimplementierungen.

Dateien

Routes/+layout.svelte

Routes/styles.css

src/app.html

Übertreffung der minimal anforderungen

Mehr als 4 DB Funktionen benutzt

Es wurden 6 DB-Funktionen mindestens einmal verwendet:

- getItems,
- getItem,
- createitem,
- updateitem,
- deleteitem,
- getLocations

Dateien:

Routes/Cart/+page.server.js

Routes/Items/+page.server.js

Routes/Itemcreate/+page.server.js

Routes/Items/[_id]/+page.server.js

Routes/Items/[_id]/create/+server.js

Routes/Items/[_id]/remove-from-cart/+server.js

Routes/Locations/+page.server.js

Routes/+page.server.js

Detail Seite Zeigt Komponenten 2 Entitäten auf

Die Produktdetailseite zeigt sowie Attribute vom Entität Items, Reviews, die einfachheitshalber in der gleichen Collection wie Items ist und Locations, welche sich in einer anderen Collection der gleichen Datenbank befindet.

Dateien:

Routes/Items/+page.server.js

Routes/Items/+page.svelte

Mehrere Formulare verschiedener typen (inkl. Multi Auswahl Feld)

Ich habe ein Formular create unter "Items/itemcreate" und ein Formular unter /[_id]/create update Formular. Das Formular unter "Items/itemcreate" erlaubt auch die Auswahl mehrerer

Dateien

Routes/Items/[_id]/create/+page.server.js

Routes/Items/[_id]/create/+server.js

Routes/Itemcreate/+page.server.js

Routes/Itemcreate/+page.svelte

Funktionen:

function handleSubmit()

M zu N Beziehung

Ich habe eine m zu n Beziehung zwischen Items und Locations. Die Locations welche bei der Item detail ansicht angezeigt werde daher via fremdschlüssel der sich in einem Array befindet Ausgelesen.

Items MongoDB abschnitt:

```
8   cart: true
9   ▼ locationsID: Array (3)
10    0: "6776a633eb3b84f6e804ee58"
11    1: "6776a633eb3b84f6e804ee59"
12    2: "6776a633eb3b84f6e804ee5a"
```

Dateien

Routes/Items/[_id] /+page.server.js

Routes/Items/[_id] /+page.server.js