

Studienarbeit

Titel: Webshop

Studiengang Informatik

An der Dualen Hochschule Baden-Württemberg

Stuttgart

von

Bela Föhrenbacher

&

Florian Kehl

Matrikelnummern: 1794796, 5848219

Betreuer: Dominik Rietz

dominik.rietz@zf.com

Zeitraum: 01.10.2021 – 10.06.2022

Abgabedatum: 10.06.2022

Abstract

Der zwischen 2010 und 2015 aufgekommene Trend, dass immer mehr Menschen immer häufiger auf den online Handel zurückgreifen, um sich zum Beispiel den Fahrtweg in die Stadt oder zum nächsten Laden zu sparen, sei es der Gemütlichkeit oder Zeitersparnis wegen. Stattdessen klagen lokale Einzelhändler vermehrt über sinkende Besucherzahlen und sind gezwungen ihre Geschäftsmodelle zu überdenken [vgl. Heinemann (2016)].

Eine der aus den Entwicklungen hervorgehenden Möglichkeiten für Einzelhändler wäre es zum Beispiel selbst einen Webshop aufzusetzen, um Kunden, die zum Onlinehandel tendieren nicht zu verlieren. Für diesen Schritt stellt sich die Frage, ob man auf bestehende Ressourcen oder Marktplätze wie Amazon oder Shopify zurückgreift oder die Entwicklung des Webshops selbst in die Hand nimmt, beziehungsweise einen Entwickler für diese beauftragt.

Die manuelle Entwicklung eines Webshops wird in der folgenden Studienarbeit durchgeführt und untersucht.

Inhaltsverzeichnis

Abstract	I
Abbildungsverzeichnis	IV
Abkürzungsverzeichnis.....	V
Ehrenwörtliche Erklärung	VI
1 Einleitung.....	1
1.1 Herausforderung.....	1
1.2 Aufgabenstellung.....	1
2 Theoretische Grundlagen.....	2
2.1 Konzeptionierung	2
2.2 Auswahl des Frameworks.....	3
2.3 Auswahl einer Datenbank	4
3 Implementation	5
3.1 Designentwurf	5
3.1.1 Grundsätzlicher Aufbau	5
3.1.2 Detailansicht.....	6
3.1.3 Nutzer Authentifizierung.....	7
3.1.4 Admin Ansicht: Inventur	8
3.1.5 Admin Ansicht: Benutzerverwaltung	9
3.1.6 Bestellprozess & Einkaufswagen.....	11

3.2	Praktische Implementierung	14
3.2.1	Einbindung der Datenbank	14
3.2.2	Grundlegendes Design	15
3.2.3	Detailansicht.....	16
3.2.4	Nutzer Authentifizierung.....	17
3.2.5	Bestellprozess & Einkaufswagen.....	19
3.2.6	Inventur	22
3.2.7	Benutzerverwaltung.....	23
3.2.8	Statusänderung bei Bestellungen	24
4	<i>Zusammenfassung und Ausblick</i>	24
4.1	Fazit & Erkenntnisse aus der Anwendung	24
4.2	Ausblick.....	25
5	<i>Quellenverzeichnis</i>	26
6	<i>Glossar.....</i>	27

Abbildungsverzeichnis

Abb. 1 Skizze der Produktansicht	5
Abb. 2 Skizze der Detailansicht eines Produkts	6
Abb. 3 Skizze der Login Seite	7
Abb. 4 Skizze der Produktübersicht aus Sicht eines Adminkontos	8
Abb. 5 Skizze der Admin Ansicht auf Benutzerkonten	9
Abb. 6 Skizze der Bearbeitungsansicht eines Benutzerprofils	10
Abb. 7 Skizze des Pop Ups zum Löschen oder Sperren eines Benutzerprofils	10
Abb. 8 Skizze zur Übersicht der zu bestellenden Artikel	11
Abb. 9 Skizze zur Adresseingabe im Bestellprozess	12
Abb. 10 Skizze zur Auswahl der Bezahlmethode	13
Abb. 11 Beispielhaftes Design der Rechnung	13
Abb. 12 Backend Code: Verbindung zur Datenbank	14
Abb. 13 Produktübersicht	15
Abb. 14 Detailansicht	16
Abb. 15 Bestätigung von Hinzufügen eines Artikels zum Warenkorb	16
Abb. 16 Anmeldeseite des Webshops	17
Abb. 17 Login Funktion im Backend	18
Abb. 18 Warenkorb mit Artikeln	19
Abb. 19 Adresseingabe	19
Abb. 20 Auswahl der Zahlungsart	20
Abb. 21 Bestellzusammenfassung	20
Abb. 22 Bestellbestätigung	21
Abb. 23 Inventurfunktion mit Administratorkonto	22
Abb. 24 Benutzerverwaltung	23

Abb. 25 Backend Funktionen zum Löschen und Sperren von Konten	23
Abb. 26 Statusänderung von Bestellungen	24

Abkürzungsverzeichnis

NoSQL	<u>N</u> o <u>S</u> QL / <u>N</u> ot <u>o</u> nly <u>S</u> QL
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
URL	<u>U</u> niform <u>R</u> esource <u>L</u> ocator

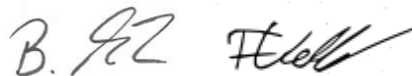
Ehrenwörtliche Erklärung

Hiermit erklären wir ehrenwörtlich, dass wir die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Aus den benutzten Quellen, direkt oder indirekt, übernommene Gedanken haben wir als solche kenntlich gemacht.

Diese Arbeit wurde bisher in gleicher oder ähnlicher Form oder auszugsweise noch keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Stuttgart, 10.06.2022

Ort, Datum



Unterschriften (Bela Föhrenbacher, Florian Kehl)

1 Einleitung

Die Folgende Studienarbeit behandelt die Erstellung eines Webshops von Grund auf, mit den im Informatik Studium erlernten Methoden. Dabei wird in der Arbeit sowohl auf die theoretischen Grundlagen, zum Beispiel zur Auswahl eines passenden Frameworks oder einer den Anforderungen entsprechenden Datenbank, als auch auf die Umsetzung der gewählten Methoden und Funktionalitäten.

1.1 Herausforderung

In den letzten Jahren zeichnete sich der Trend ab, dass die Menschen immer häufiger auf den Onlinehandel, statt den lokalen Einzelhandel zurückgreifen [vgl. Heinemann (2016)]. Um dieser Entwicklung entgegenzuwirken, sollten auch kleine Einzelhändler in Erwägung ziehen, eine online Präsenz in Form eines Webshops aufzubauen.

1.2 Aufgabenstellung

Aufgabe für die Studienarbeit ist das Erstellen eines Webshops. Dieser soll wie bekannte Stores von Amazon und Co. funktionieren. Benutzende sollen Warenkörbe anlegen, Benutzerkonten erstellen und Bestellungen verwalten können. Darüber hinaus soll es bei vergessenem Passwort möglich sein, wieder Zugriff auf sein Benutzerkonto zu bekommen, ebenso wie die Möglichkeit sein Passwort zu ändern.

Der Webshop soll eine Produktübersicht mit Bildern und Produktnamen zeigen und die Möglichkeit zur Filterung und Suche nach Produktgruppen geben. Außerdem soll zu den Produkten eine Detailansicht mit Beschreibung und zum Beispiel einer Bestandsanzeige existieren.

Ein Bestellprozess soll ebenfalls theoretisch umgesetzt werden. Hierbei wird die Bezahlungsmethode zu „auf Rechnung“ reduziert, um unnötige Finanztransaktionen und die Implementation von Services wie PayPal zu vermeiden. Die bestellende Person soll jedoch eine Mail mit Rechnung als Bestellbestätigung erhalten können.

Zusätzlich soll auch ein Administrationskonto eingerichtet werden, über welches unter anderem neue Produkte eingestellt werden, der Bestand angepasst (Inventur) und Benutzende verwaltet werden können.

2 Theoretische Grundlagen

Das folgende Kapitel dient der Formulierung der Grundlagen des zu erstellenden Webshops. Diese wurden in drei Unterpunkte aufgeteilt:

- Konzeptionierung, z.B. das Festlegen der Komponenten und Anforderungen
- Auswahl des Frameworks für die Webseite
- Auswahl einer Datenbank zum Speichern von Produkten, Nutzerdaten, etc.

Auf diese vordefinierten Punkte wird nachfolgend genauer eingegangen, um die Entscheidungsprozesse verständlich zu machen.

2.1 Konzeptionierung

Der erste Schritt ist die grundlegende Konzeptionierung der Anforderungen und Komponenten für den Webshop.

Die Anforderungen wurden vom Product Owner im initialen Meeting wie folgt formuliert:

Der Webshop soll über die typischen Funktionen bekannter Onlinemarktplätze verfügen. Benutzende sollen die verfügbaren Waren einsehen können, zu einem Warenkorb hinzufügen und bestellen können. Darüber hinaus soll es möglich sein ein Benutzerkonto anzulegen, mit welchem dann getätigte Bestellungen verknüpft werden sollen. Benutzende sollen außerdem ihr Benutzerkonto verwalten können. Dazu gehören unter anderem die Anpassung von Adressdaten oder das Ändern des Passworts.

Loggt man sich mit einem Administratorkonto ein, soll es möglich sein, Produkte zum Inventar hinzuzufügen oder zu entfernen, den Lagerbestand anzupassen (Inventur) und Benutzerkonten zu verwalten, wie zum Beispiel das Sperren von Benutzerkonten oder zurücksetzen von Passwörtern.

Die verfügbaren Produkte sollen in einer Übersicht angezeigt werden, mit der zusätzlichen Möglichkeit der Stichwortsuche, der Filterung und einer Detailansicht.

2.2 Auswahl des Frameworks

Zur Erstellung von Webseiten stehen verschiedene Möglichkeiten zur Auswahl. Eine Webseite kann in schlichtem HTML geschrieben werden, mit CSS und Skriptsprachen können Design und Funktionalitäten hinzugefügt werden. Es gibt „Baukastensysteme“, in welchen man sich die Webseite grafisch zusammenstellen kann, ohne sich mit dem Code beschäftigen zu müssen, Content-Management-Systeme, wie zum Beispiel WordPress oder Frameworks wie AngularJS oder Vue, die einen Rahmen für eine individualisierbare Webseite bilden, durch das Bereitstellen von Webseite-Elementen, wie fertigen Navigationsleisten, Bildergalerien oder verschiedenen Listen- und Galerieansichten für unterschiedliche Inhalte. Eine Webseite selbst zu programmieren, sei es mit Frameworks oder in schlichtem HTML bietet das größte Spektrum an Flexibilität, ist jedoch mit hoher Komplexität verbunden, im Vergleich zu den anderen aufgeführten Optionen [Schmidt (2022)].

Im Falle des in der vorliegenden Arbeit beschriebenen Webshop wurde sich für Angular entschieden. Diese Entscheidung hat verschiedene Gründe. Der trivialste Grund ist, dass das Entwicklerteam bereits Erfahrungen mit Angular gesammelt hat und somit keine neuen Tools erlernt werden müssen. Darüber hinaus gibt es mit Nebular ein sehr flexibles UI-Kit für Angular mit guten Designvorlagen für Webseiten-Elemente wie *Navbars* oder Produktlisten. Außerdem bietet Nebular auch eine *Auth*-Komponente zur Integration von Benutzerauthentifizierung.

2.3 Auswahl einer Datenbank

Um Kundenkonten oder Artikel verarbeiten zu können benötigt der Webshop eine Datenbank im Hintergrund, in welcher die Informationen gespeichert werden. Da es unterschiedliche Datenbankmodelle zur Auswahl gibt, muss eine passende Wahl getroffen werden.

Grundlegend wird in vier Datenbankmodelle unterschieden, relationale Datenbanken, objektorientierte Datenbanken, hierarchische Datenbanken und netzwerkartige Datenbanken [vgl. Schicker (2017)]. Darüber hinaus werden seit einigen Jahren solche Datenbanken, die nicht (oder nicht nur) auf SQL basieren in die Kategorie der NoSQL Datenbanken eingeteilt. Dazu zählen Key/Value und dokumentenbasierte Modelle, Spaltenorientierte Modelle und Graph-Datenbanken [vgl. Schicker (2017)].

Außerdem können Datenbanken hinsichtlich des CAP Theorems untersucht werden. Das CAP Theorem sagt aus, dass Verteilte Systeme, wie zum Beispiel Datenbanken nur zwei von drei angestrebten Charakteristiken erfüllen können. Diese sind Konsistenz (Consistency), Verfügbarkeit (Availability) und Partitionstoleranz (Partition Tolerance) [vgl. IBM (2019)]. Abhängig von den Anforderungen an die verwendete Datenbank, ist es möglich, dass die Auswahl anhand der von einer Datenbank erfüllten Charakteristika eingeschränkt wird.

Zur näheren Auswahl standen MySQL, PostgreSQL und MongoDB, da zu diesen bereits Vorkenntnisse aus Vorlesungen und praktischer Umsetzung vorhanden waren. Die Entscheidung ist zugunsten von MongoDB gefallen, da für Dokumentenbasierte Datenbanken keine Schemata vordefiniert werden müssen und alle Daten im JSON Format gespeichert werden können. Darüber hinaus kann in der MongoDB Cloud ein kostenloses Cluster aufgesetzt werden, dessen Kapazitäten für den Webshop ausreichend sind. MongoDB erfüllt Konsistenz und Partitionstoleranz (CP) im CAP Theorem [vgl. IBM (2019)], jedoch wurde für den ersten Entwurf des Webshops eine tiefergehende Betrachtung vernachlässigt, da der Fokus mehr auf einem praktischen Ergebnis liegt und keiner theoretischen Analyse.

3 Implementation

Der folgende Abschnitt der vorliegenden Arbeit beschreibt das Design und die Implementierung des Webshops. Die Beschreibung wird mit Entwurfsskizzen, Screenshots und Ausschnitten aus dem Code unterstützt.

3.1 Designentwurf

Die kommenden Unterkapitel enthalten die Skizzen mit Designentwürfen zu den verschiedenen Funktionalitäten des Webshops.

3.1.1 Grundsätzlicher Aufbau

Das grundlegende Design des Webshops soll sich an bekannten Stores anderer Onlinehändler orientieren. Es soll eine Kopfzeile geben, mit „Suchen“-Funktion, einer Verlinkung zum Benutzerprofil und dem Warenkorb. In einer Navigationsleiste an der Seite soll gezielt nach Kategorien gesucht werden können und die angebotenen Produkte sollen in Kacheln angezeigt werden.

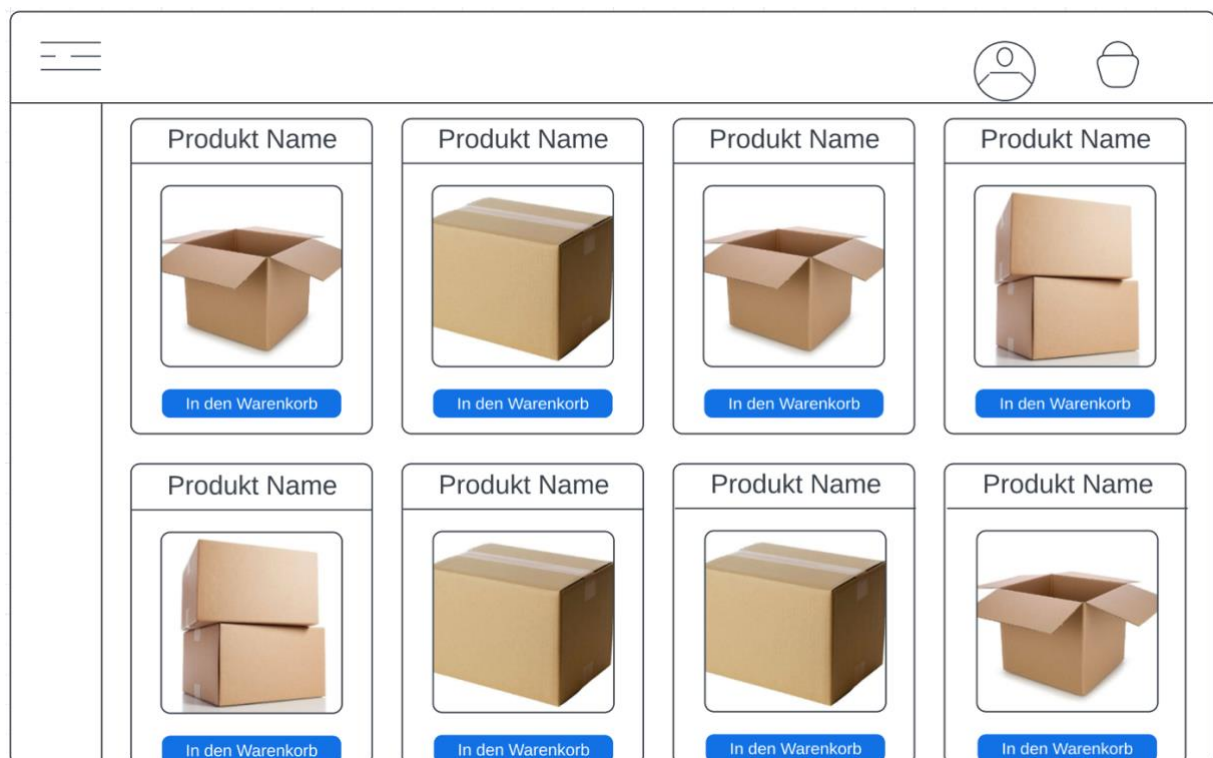


Abb. 1 Skizze der Produktansicht

Die Skizze in Abb. 1 zeigt grob den geplanten Aufbau des Webshops. Oben rechts findet sich der Warenkorb und das Icon des Benutzerprofils, oben links das Icon, um die Sidebar zu öffnen. Im Vordergrund stehen die Produktkacheln, mit Produktnamen, Produktbild und einem Knopf, um diese in den Warenkorb zu tun. Wird eine Produktkachel angeklickt, soll sich eine detailliertere Ansicht öffnen.

3.1.2 Detailansicht

Die Detailansicht für Produkte soll dann wie in folgender Skizze dargestellt aussehen und weitere Produktbilder und Informationen enthalten:

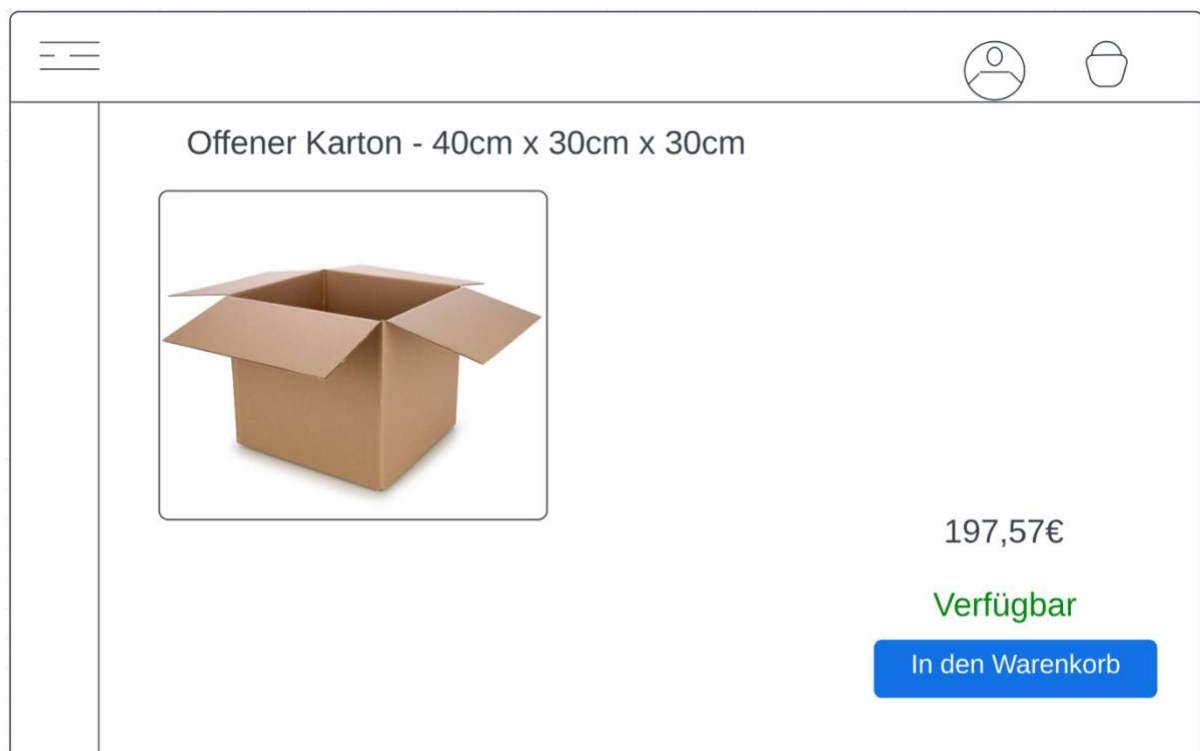


Abb. 2 Skizze der Detailansicht eines Produkts

Hier sollen dann zum Beispiel genauere Verfügbarkeitsinformationen oder Lieferinformationen stehen und zum Beispiel Kundenbewertungen eingestellt oder gelesen werden. Es soll außerdem möglich sein über die Detailansicht mehr als nur eine Einheit eines Produktes auf einmal in den Warenkorb legen zu können.

3.1.3 Nutzer Authentifizierung

Benutzenden des Webshops soll es möglich sein, sich mit einem Benutzerprofil anzumelden. In diesem soll der Warenkorb hinterlegt und Informationen über vergangene Bestellungen archiviert werden.

The sketch shows a web browser window. The header bar contains a hamburger menu icon on the left and two icons on the right: a person icon and a shopping basket icon. The main content area is centered and contains the following elements:

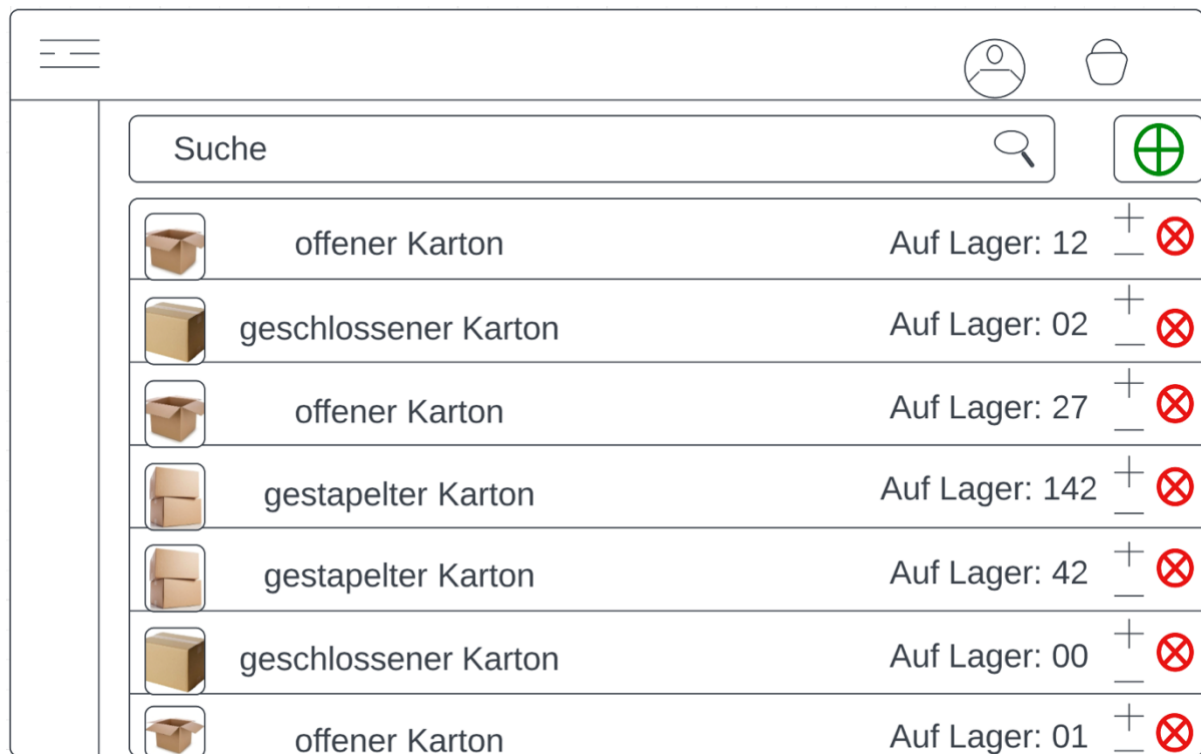
- An input field labeled "e-mail Adresse".
- An input field labeled "Passwort".
- A blue button labeled "Login".
- Below the button, the text "Noch Kein Account? [Registrieren](#)".
- Below that, the text "Passwort vergessen? [Zurücksetzen](#)".

Abb. 3 Skizze der Login Seite

Abb. 3 zeigt eine Entwurfsskizze, wie die Login Seite ungefähr aussehen soll. Die Anmeldung soll mit der E-Mail-Adresse und einem Passwort möglich sein. Existiert für eine Person noch keinen Account soll diese sich über den Link zur Registrierungsseite einen erstellen können und wurde das Passwort vergessen ist ein Link vorhanden, um den Prozess zum Zurücksetzen dieses zu starten.

3.1.4 Admin Ansicht: Inventur

Als Admin eingeloggt soll die Produktübersicht etwas anders gestaltet sein. In einer Listenansicht sollen alle Produkte zu sehen sein, über eine Suche sollen einzelne Produkte zu finden sein. Es soll direkt ersichtlich sein, wie viele Produkte auf Lager sind und die Anzahl muss erhöht oder verringert werden können. Außerdem müssen Produkte zu löschen oder neu anzulegen sein.



The sketch shows a web interface for an admin inventory view. At the top, there is a header bar with a hamburger menu icon on the left, and user and basket icons on the right. Below the header is a search bar with the placeholder text 'Suche', a magnifying glass icon, and a green plus icon in a circle. The main content area is a table with seven rows, each representing a product. Each row contains a product icon (a cardboard box), the product name, the stock quantity, and two action icons: a plus/minus control and a red delete 'X' icon.

Icon	Produktname	Auf Lager	Steuerung	Löschen
	offener Karton	12	+ -	✗
	geschlossener Karton	02	+ -	✗
	offener Karton	27	+ -	✗
	gestapelter Karton	142	+ -	✗
	gestapelter Karton	42	+ -	✗
	geschlossener Karton	00	+ -	✗
	offener Karton	01	+ -	✗

Abb. 4 Skizze der Produktübersicht aus Sicht eines Adminkontos

Abb. 4 zeigt eine grobe Skizze, wie diese Ansicht umzusetzen sein könnte. Die notwendigen Funktionen zur Inventur sind vorhanden.

3.1.5 Admin Ansicht: Benutzerverwaltung

Über die Möglichkeit das Inventar zu bearbeiten hinweg, soll von einem Administratorkonto aus auch die Benutzerverwaltung möglich sein. Diese soll Funktionen, wie das Zurücksetzen von Passwörtern, Bearbeiten von Profilinformationen und das Sperren oder Löschen von Benutzerprofilen, umfassen.

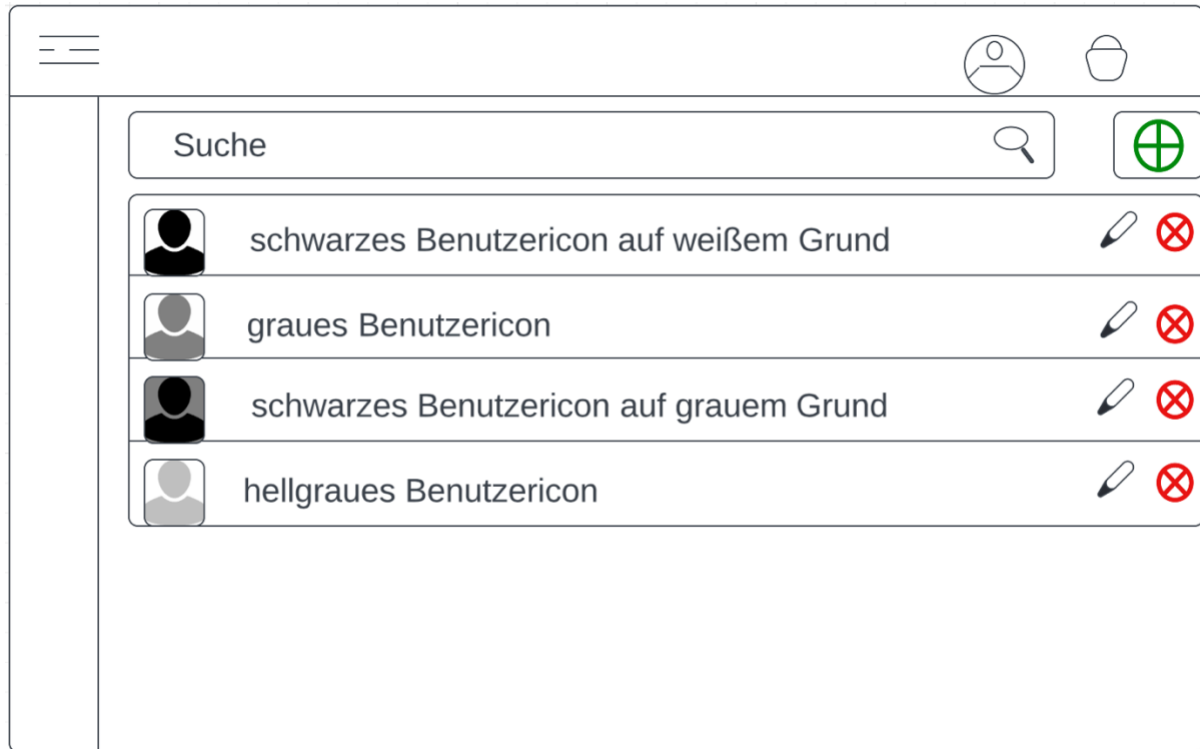


Abb. 5 Skizze der Admin Ansicht auf Benutzerkonten

Wird der Stift angeklickt soll sich ein Fenster zur Bearbeitung des ausgewählten Benutzerprofils öffnen. Dieses soll einen Knopf zum Senden einer Wiederherstellungsemail für das Passwort und ein Formular zur Bearbeitung enthalten.

Wird das rote X angeklickt, soll ein Pop Up erscheinen und fragen, ob das Benutzerprofil gelöscht oder nur vorübergehend gesperrt werden soll.

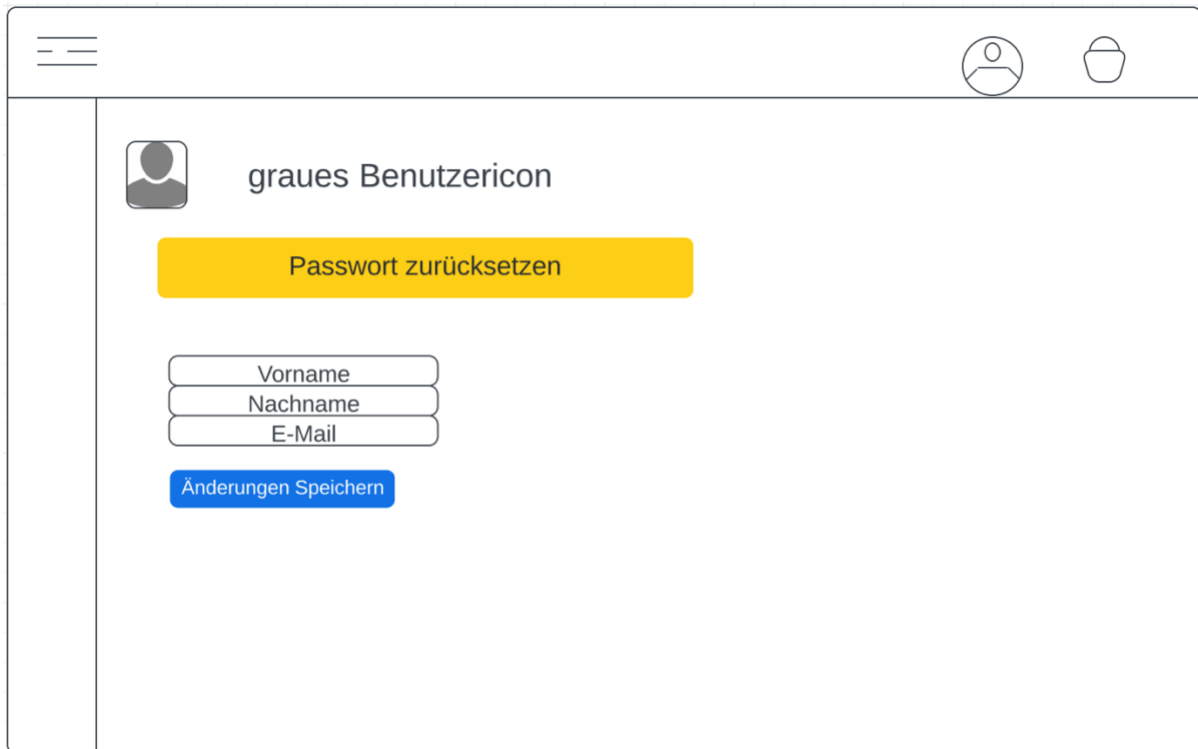


Abb. 6 Skizze der Bearbeitungsansicht eines Benutzerprofils

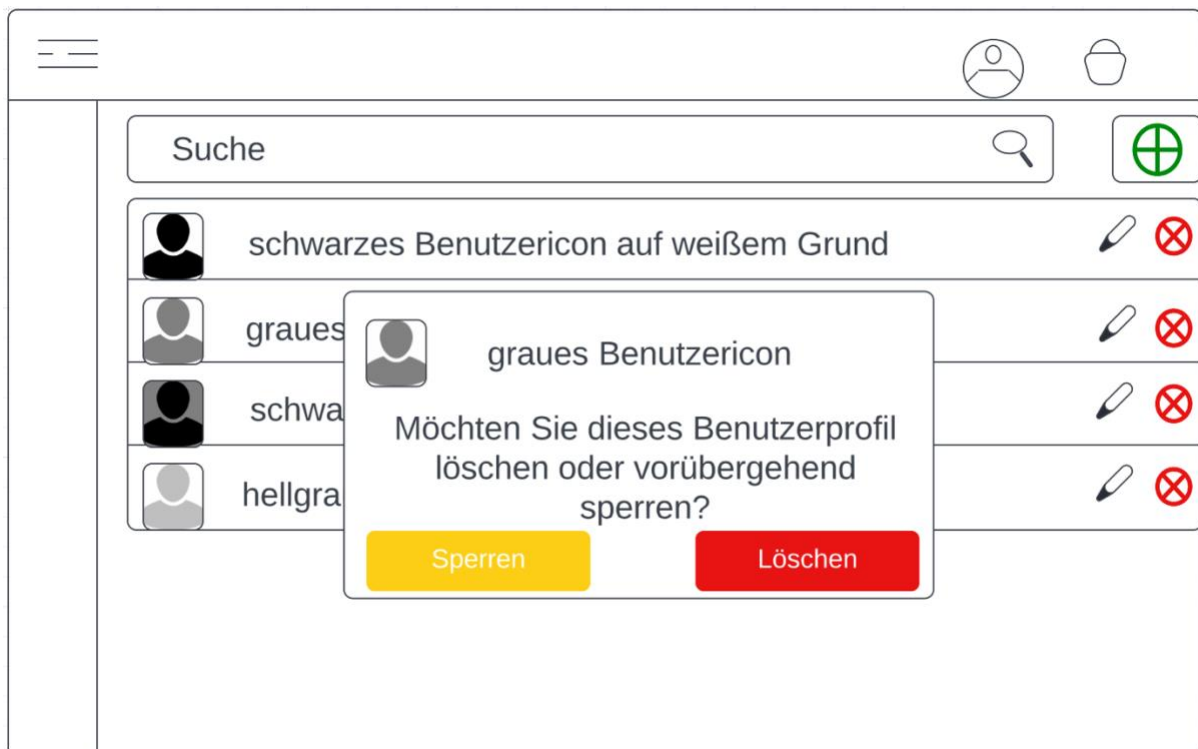


Abb. 7 Skizze des Pop Ups zum Löschen oder Sperren eines Benutzerprofils

3.1.6 Bestellprozess & Einkaufswagen

Artikel, die eine benutzende Person kaufen möchte, sollen wie in jedem Webshop üblich in einen virtuellen Einkaufswagen getan werden können. Dieser soll mit dem Benutzerprofil verknüpft werden, damit Warenkörbe auch über eine Browser Session hinaus gespeichert werden können. Darüber hinaus wäre die Möglichkeit den Warenkorb im Session Storage oder als Cookies zu speichern eine Option, die in Betracht gezogen werden könnte, damit auch nicht angemeldete Personen einen Einkaufswagen anlegen könnten.

Der Ablauf des Bestellprozesses sollte im Idealfall unabhängig von der Existenz eines Benutzerkontos funktionieren. Der Warenkorb soll dafür eine Check-out Funktion erhalten, die den Bestellprozess mit den im Warenkorb befindlichen Artikeln startet.

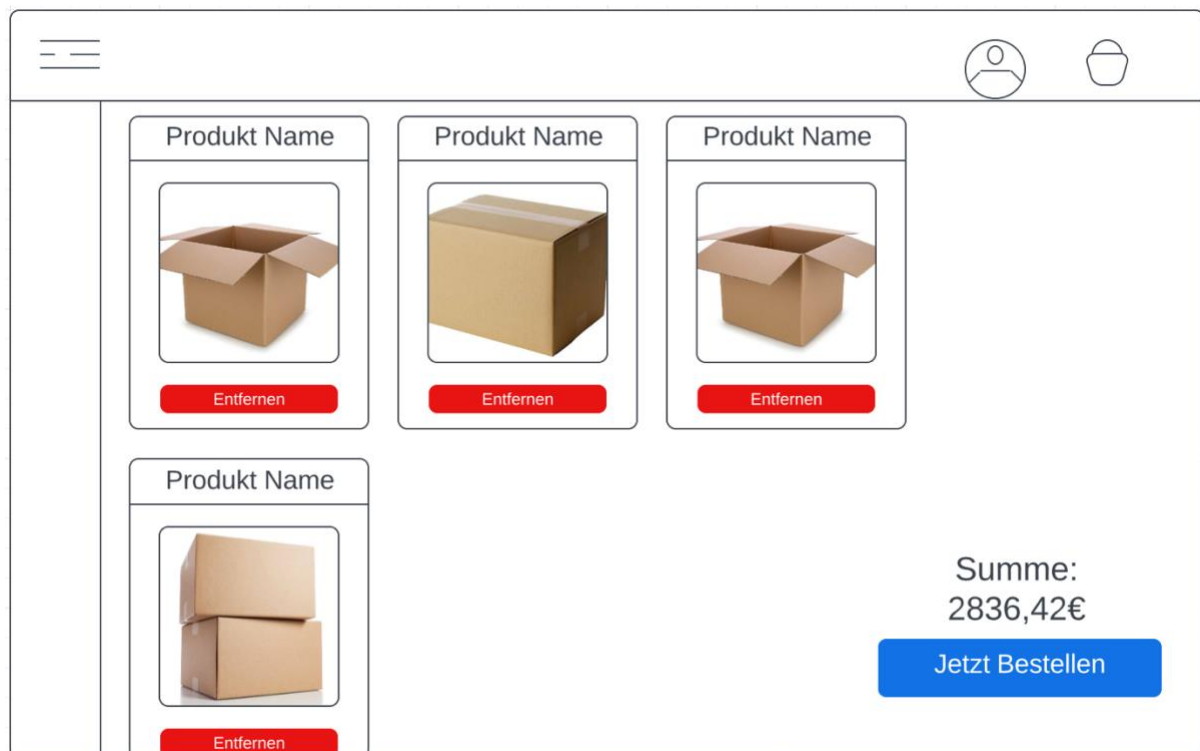


Abb. 8 Skizze zur Übersicht der zu bestellenden Artikel

Hier soll wie in Abb. 8 dargestellt eine Übersicht der im Warenkorb befindlichen Artikel erscheinen und die Möglichkeit gegeben werden die Anzahl noch einmal zu verändern oder einen Artikel zu entfernen. Darüber hinaus soll die Gesamtsumme (vor Versandkosten) angezeigt werden. Über den Knopf „Jetzt Bestellen“ kann der Bestellprozess fortgeführt werden.

Es folgt die in Abb. 9 skizzierte Eingabe von Rechnungs- und Lieferadresse. Die Lieferadresse ist hierbei optional und muss nur angegeben werden, wenn sich diese von der Rechnungsadresse unterscheidet.

The sketch shows a web form titled 'Adresseingabe:'. It contains two sections: 'Rechnungsadresse:' and an optional 'Abweichende Lieferadresse'. Each section has input fields for 'Vorname', 'Nachname', 'Straße' (with a 'Nr.' field), 'PLZ', and 'Wohnort'. The 'Abweichende Lieferadresse' section is preceded by a checkbox.



Abb. 9 Skizze zur Adresseingabe im Bestellprozess

Nach der Eingabe von Liefer- und Rechnungsadresse erfolgt die Auswahl der Bezahlmethode, bevor eine finale Überprüfung der zu bestellenden Artikel durchgeführt wird.

Als Bezahlmethode soll, wie in Abb. 10 skizziert nur der Kauf auf Rechnung angeboten werden, da somit keine Zahldienstleister eingebunden werden müssen. Damit die Seite nicht zu leer aussieht sind die Auswahlmöglichkeiten für ein SEPA Lastschriftmandat und die Bezahlung mit PayPal bereits vorhanden, jedoch aktuell nicht auswählbar. Die Implementierung der Zahlungsdienstleister wäre eine mögliche Aufgabe, die in Anschluss an die Grundlegende Entwicklung des Webshops aufgenommen werden könnte.

Bezahlungsmethode:

☐ Kauf auf Rechnung

☐  SEPA Lastschriftmandat (zurzeit nicht möglich)
 ☐  Bezahlung mit PayPal (zurzeit nicht möglich)

*Aus Entwicklungsgründen ist Zahlung aktuell nur auf Rechnung möglich.

Abb. 10 Skizze zur Auswahl der Bezahlungsmethode

Die Rechnung soll in HTML generiert werden und wenn möglich darüber hinaus als PDF per Mail an die bestellende Person zugestellt werden.

Verpackungszubehör.de

Rechnungsadresse:

Max Mustermann

Gartenstraße 12

Berlin

Hallo Max,

nachfolgend findest Du die Bestell- und Lieferinformationen zu deiner Bestellung mit der Bestellnummer 123456

Produkte:

Name	Menge	Stückpreis	Summe
Produkt 1	1	42	42
Produkt 2	3	33	99
Produkt 3	2	4	8
Total:			149,-

Lieferadresse:

Max Mustermann

Gartenstraße 12

Berlin

Zahlung an:

Bela & Flo

IBAN: DE12 3456 7890 9876 43

BANKDE 33XXX

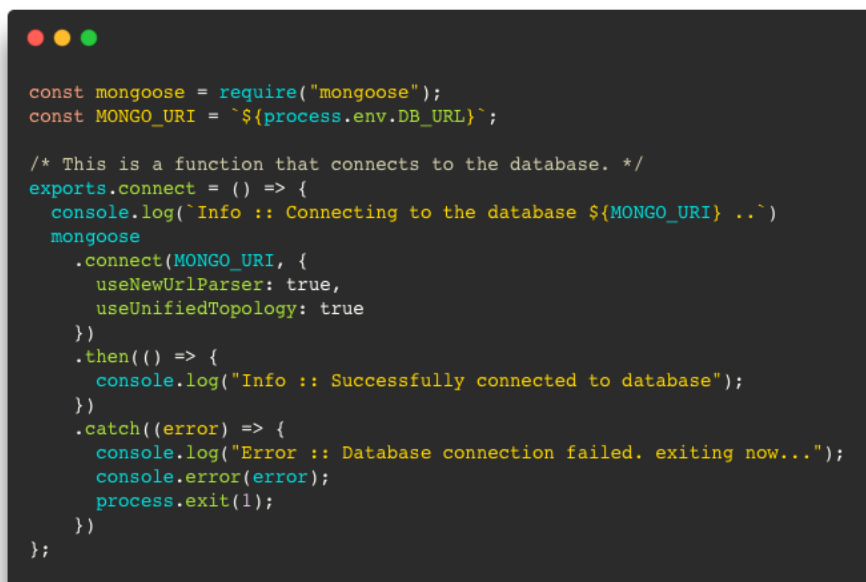
Abb. 11 Beispielhaftes Design der Rechnung

3.2 Praktische Implementierung

Die folgenden Unterkapitel beschreiben die tatsächliche Umsetzung der zuvor konzeptionierten Funktionalitäten und des geplanten Designs. Überwiegend wurde sich hierbei an den zuvor erstellten Skizzen orientiert.

3.2.1 Einbindung der Datenbank

Der folgende Codeausschnitt zeigt wie in Node.js die Verbindung zur Datenbank hergestellt wurde. Dafür wurde das Mongoose Package verwendet, welches mit der `connect()` Funktion eine Datenbankverbindung aufbaut [vgl. TutorialKart (o. J.)].



```
const mongoose = require("mongoose");
const MONGO_URI = `${process.env.DB_URL}`;

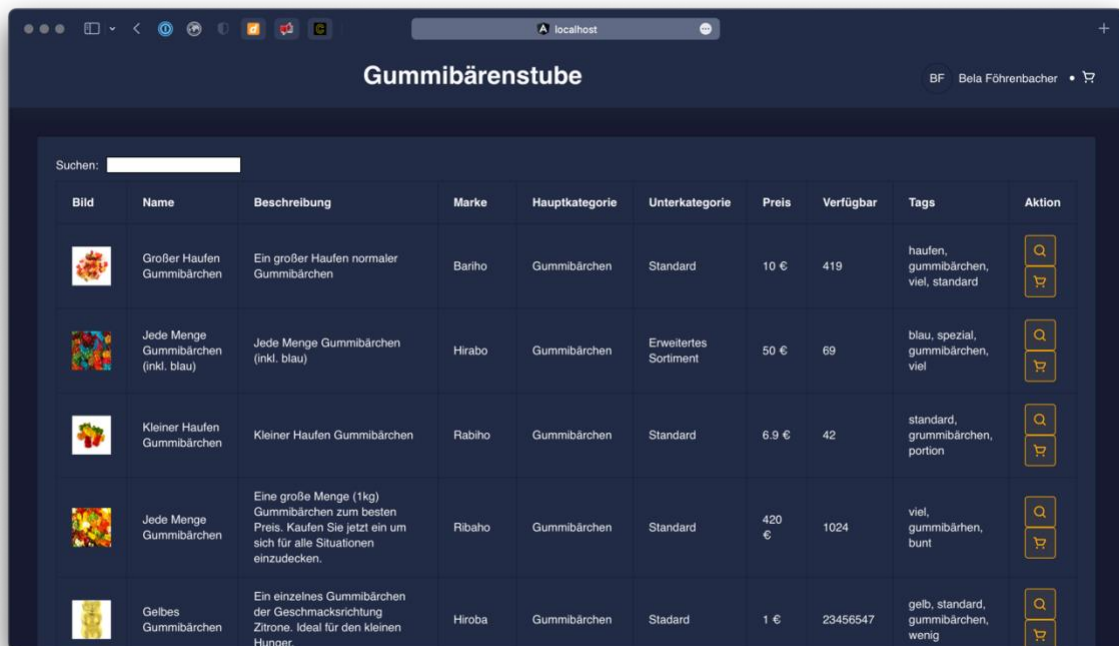
/* This is a function that connects to the database. */
exports.connect = () => {
  console.log(`Info :: Connecting to the database ${MONGO_URI} ..`)
  mongoose
    .connect(MONGO_URI, {
      useNewUrlParser: true,
      useUnifiedTopology: true
    })
    .then(() => {
      console.log("Info :: Successfully connected to database");
    })
    .catch((error) => {
      console.log("Error :: Database connection failed. exiting now...");
      console.error(error);
      process.exit(1);
    })
};
```

Abb. 12 Backend Code: Verbindung zur Datenbank

Diese Datenbankverbindung kann dann für Datenbankabfragen verwendet werden, wie zum Beispiel das Erstellen eines Warenkorbes oder die Veränderung des Inventars.

3.2.2 Grundlegendes Design

Das grundlegende Design des Webshops wurde entgegen der ursprünglichen Planung nicht in einer Kachelansicht umgesetzt, sondern in einer Listenansicht. Auch diese Methode ist von großen online Händlern bekannt und wurde als etwas übersichtlicher empfunden. Darüber hinaus war es in der Umsetzung etwas einfacher zu gestalten.



The screenshot shows a web browser window with the address bar displaying 'localhost'. The page title is 'Gummibärenstube'. In the top right corner, there is a user profile 'BF Bela Föhrenbacher' and a shopping cart icon. Below the header, there is a search bar labeled 'Suchen:'. The main content area displays a table of products with the following columns: Bild, Name, Beschreibung, Marke, Hauptkategorie, Unterkategorie, Preis, Verfügbar, Tags, and Aktion. The table contains five rows of product data.
















Bild	Name	Beschreibung	Marke	Hauptkategorie	Unterkategorie	Preis	Verfügbar	Tags	Aktion
	Großer Haufen Gummibärchen	Ein großer Haufen normaler Gummibärchen	Bariho	Gummibärchen	Standard	10 €	419	haufen, gummibärchen, viel, standard	 
	Jede Menge Gummibärchen (inkl. blau)	Jede Menge Gummibärchen (inkl. blau)	Hirabo	Gummibärchen	Erweitertes Sortiment	50 €	69	blau, spezial, gummibärchen, viel	 
	Kleiner Haufen Gummibärchen	Kleiner Haufen Gummibärchen	Rabiho	Gummibärchen	Standard	6.9 €	42	standard, gummibärchen, portion	 
	Jede Menge Gummibärchen	Eine große Menge (1kg) Gummibärchen zum besten Preis. Kaufen Sie jetzt ein um sich für alle Situationen einzudecken.	Ribaho	Gummibärchen	Standard	420 €	1024	viel, gummibärchen, bunt	 
	Gelbes Gummibärchen	Ein einzelnes Gummibärchen der Geschmacksrichtung Zitrone. Ideal für den kleinen Hunger.	Hiroba	Gummibärchen	Standard	1 €	23456547	gelb, standard, gummibärchen, wenig	 

Abb. 13 Produktübersicht

Als Farbschema wurde für Version Eins des Webshops eines der von Nebular bereitgestellten Themes verwendet, welches jedoch für eine spätere Implementierung thematisch an die verkauften Artikel angepasst werden könnte.

Der Webshop hat eine Kopfzeile, und die Produktübersicht wie in Abb. 13 zu sehen als Hauptelement. Auf die *Navbar* aus der ursprünglichen Planung wurde für den ersten Entwurf des Webshops verzichtet. Diese könnte in einer späteren Version ergänzt werden und Funktionen wie detaillierte Filterung enthalten. Die Produkte werden in einer Liste dargestellt, mit der Option diese direkt in den Warenkorb legen zu können oder eine Detailansicht zu öffnen. Außerdem wird zu jedem Produkt ein Bild angezeigt. Die Bilder werden als *base64* in der Datenbank gespeichert. Die Alternative dazu wäre gewesen, die Bilder zu Beispiel in einem *S3 Bucket* zu

speichern und in der Datenbank zu verlinken, diese Lösung wäre jedoch aufwendiger gewesen und als mögliche zukünftige Anforderung definiert.

3.2.3 Detailansicht

Auch die Detailansicht einzelner Produkte orientiert sich was das Design betrifft an bekannten Webshops.

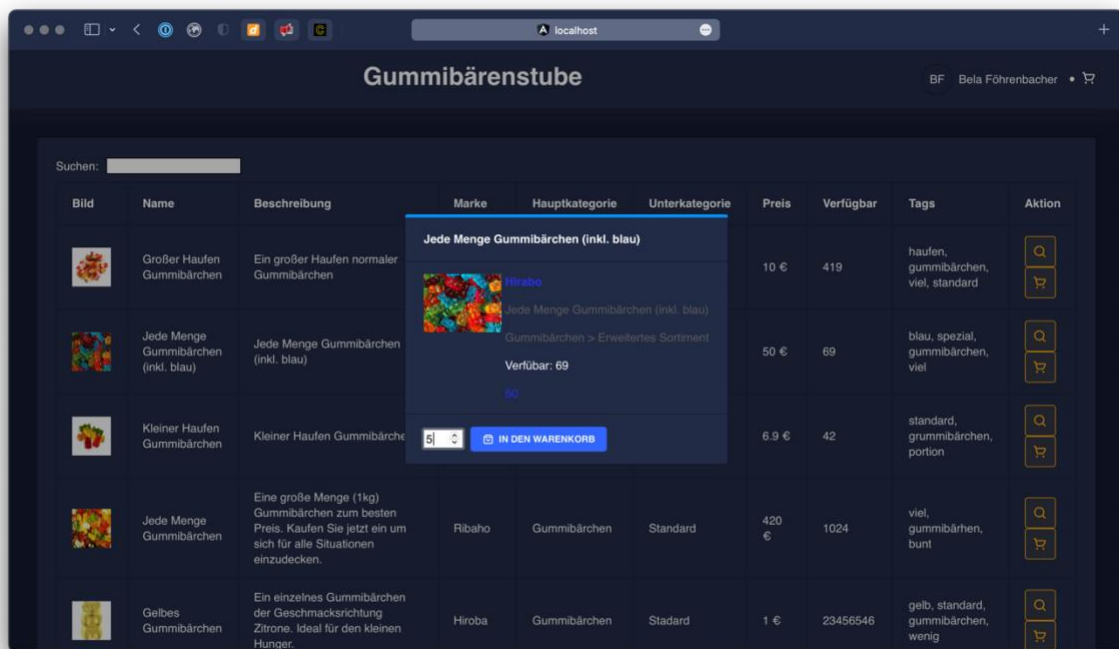


Abb. 14 Detailansicht

Benutzende können hier weitere Details einsehen und mehr als eine Einheit eines Produktes auf einmal in den Warenkorb legen.

Wurde ein Artikel in den Warenkorb gelegt, wird dies auch mit einem Hinweis quittiert:

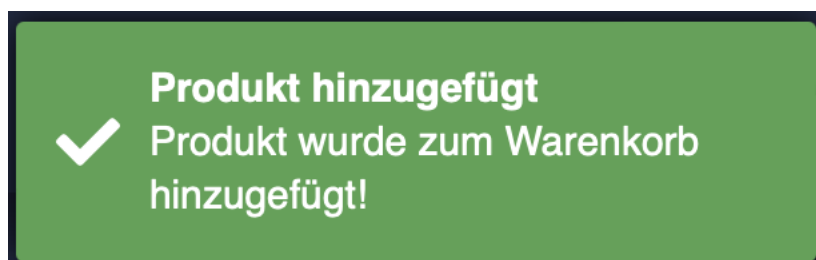


Abb. 15 Bestätigung von Hinzufügen eines Artikels zum Warenkorb

3.2.4 Nutzer Authentifizierung

Die Funktionalität zur Nutzer-Authentifizierung ist in Nebular mit der NbLoginComponent und NbRegisterComponent bereits enthalten und kann somit recht einfach integriert werden. Unterschieden werden soll zwischen Benutzenden ohne Konto, Benutzenden mit Benutzerkonto und Admins, welche Lagerbestände verändern und Produkte hinzufügen oder löschen können sollen. Dafür wird im Backend zusätzlich zur Authentifizierung mithilfe des *AuthGuards* auch ein Rollen Schutz (*RoleGuard*) hinterlegt, welcher beim Login prüft, ob eine Rolle mit entsprechenden Rechten vorhanden ist.

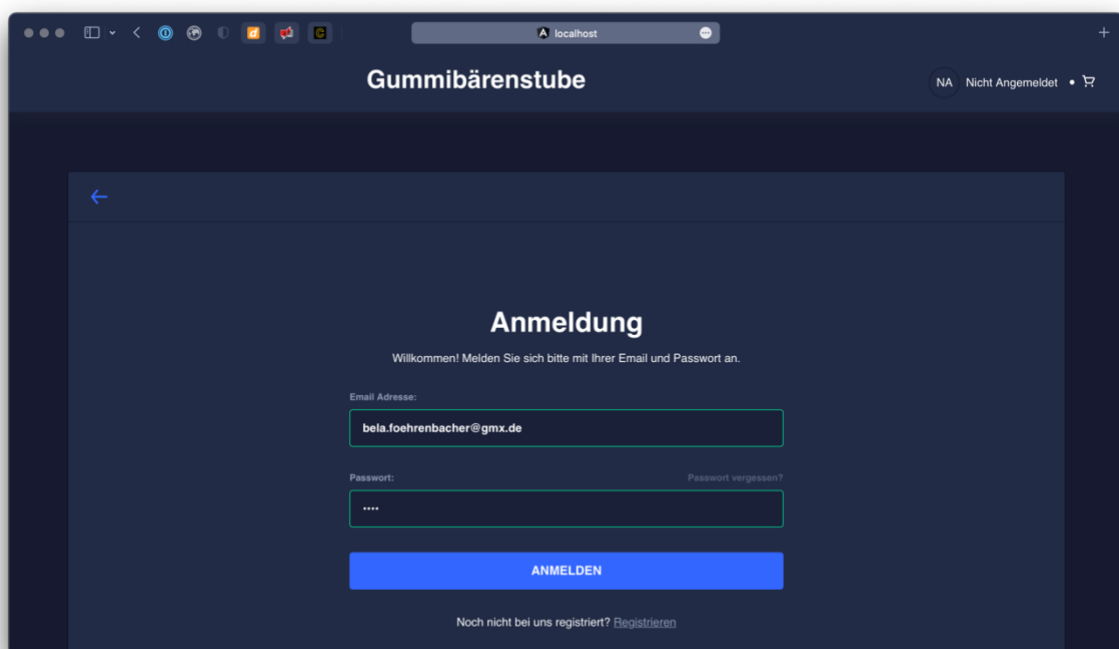


Abb. 16 Anmeldeseite des Webshops

Registrieren können sich Benutzende mit der Angabe von Namen und Vorname zur Anzeige im Profil, einer E-Mail-Adresse und einem Passwort. Zum Login werden dann die E-Mail und das Passwort benötigt. Es ist außerdem möglich das Passwort zurücksetzen zu lassen, für den Fall, dass dieses vergessen wurde.

Folgender Codeausschnitt zeigt, wie der Login im Backend umgesetzt wurde:

A screenshot of a code editor with a dark background and light-colored text. The code is written in JavaScript/Node.js and represents a login route. It includes comments, variable declarations, conditional checks for email and password, user lookup, password comparison, JWT token generation, and response handling for success and errors. The code is enclosed in a function that is called by the router.

```
/* This is the route for logging in a user. */
router.post("/login", bodyParser, async (req, res) => {

  try {
    const { email, password } = req.body;

    if (!(email && password)) {
      res.status(400).send("All input is required");
    }
    const user = await User.findOne({ email });

    if (user && (await bcrypt.compare(password, user.password))) {
      const token = jwt.sign(
        {
          _id: user._id,
          email,
          name: user.name,
          surname: user.surname,
          locked: user.locked,
          locked_message: user.locked_message,
          roles: user.roles
        }, process.env.TOKEN_KEY, {
          expiresIn: "5h",
        }
      );

      user.token = token;

      return res.status(200).json(user);
    }
    return res.status(400).send("Invalid Credentials");
  } catch (err) {
    return res.status(500).send(err);
  }
})
```

Abb. 17 Login Funktion im Backend

Über die API wird ein URL-Pfad zum Login bereitgestellt, über welchen die in Abb. 17 gezeigt Funktion aufgerufen werden kann. Hier wird dann geprüft, ob E-Mail und Passwort in der eingegebenen Kombination existieren oder ob ein Konto gesperrt wurde.

3.2.5 Bestellprozess & Einkaufswagen

Hat die benutzende Person ihre Artikelauswahl getroffen, ist diese im Warenkorb einsehbar. Dort kann die Auswahl auch noch einmal bearbeitet werden und die Bestellsumme wird angezeigt.

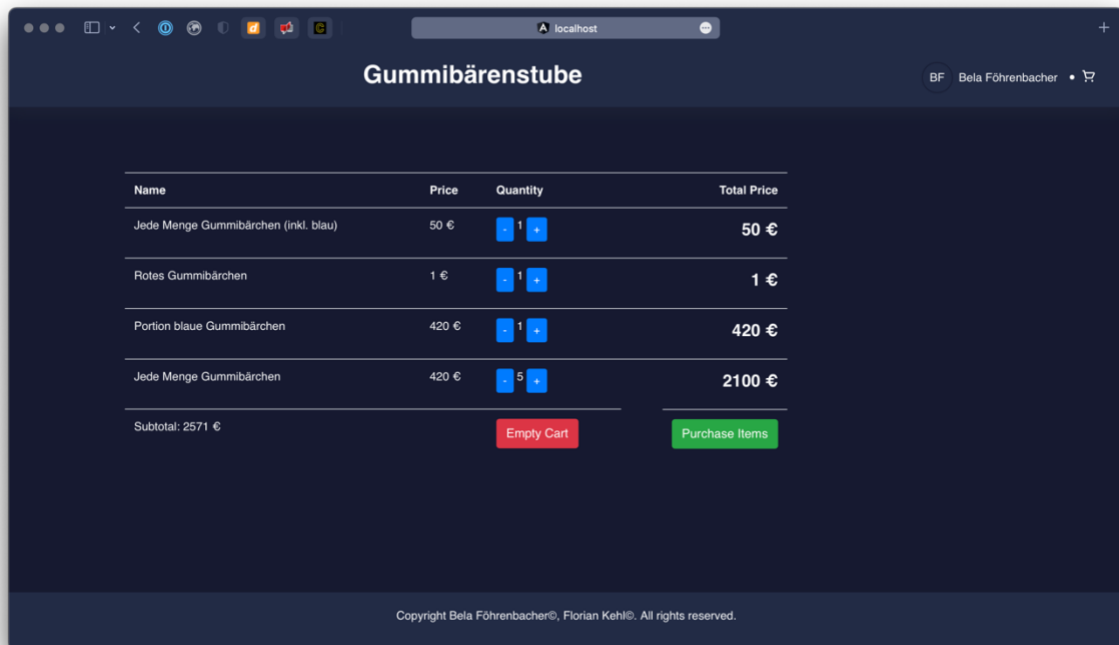


Abb. 18 Warenkorb mit Artikeln

Wird auf *Purchase Items* geklickt, wird der Bestellprozess eingeleitet. Dazu öffnet sich ein Pop-Up für die Eingabe der Bestellinformationen.

The screenshot shows a mobile app interface for the checkout process. At the top, there are three steps: 1. Adresse, 2. Zahlungsmittel, and 3. Bestätigung. The first step, 'Adresse', is active. Below it, there is a section titled 'Lieferadresse' with several input fields: 'Land' (with a dropdown menu showing 'Deutschland'), 'PLZ' (with the value '13127' and a location icon), 'Ort' (with a dropdown menu showing 'Ort'), 'Strasse' (with a dropdown menu showing 'Strasse'), 'Hausnummer' (with a dropdown menu showing 'Hausnumm'), and 'Addresszusatz' (with a dropdown menu showing 'Addresszu').

Abb. 19 Adresseingabe

Hier werden benutzende mit einem *Stepper* durch die Eingabe der Informationen wie zum Beispiel Liefer- und Rechnungsadresse oder der Zahlungsart geleitet.

Abb. 20 Auswahl der Zahlungsart

Nach der Auswahl der Bezahlmethode wird eine Rechnung in HTML generiert und als Bestellzusammenfassung angezeigt.

Gummibärenstube.de

Rechnungsadresse:

Bela FÄhnbacher
Im Biengarten 2
56368 Katzenelnbogen

Hallo Bela FÄhnbacher,
nachfolgend die Rechnung für deine Bestellung bei Gummibärenstube.de.
Bestellnummer: 62a3335f50354c3fedb46477

Produkte:

Name	Menge	Stückpreis	Summe
Jede Menge Gummibärchen (inkl. blau)	1	50 €	50 €
Rotes Gummibärchen	1	1 €	1 €
Portion blaue Gummibärchen	1	420 €	420 €
Jede Menge Gummibärchen	5	420 €	2100 €
		Total:	2571 €

Lieferadresse:

Abb. 21 Bestellzusammenfassung

Zuletzt erhält die benutzende Person eine Bestätigung über ihre Bestellung sowohl in dem Pop-Up als auch eine Bestätigungsemail an, die im Profil hinterlegt E-Mail-Adresse.



Abb. 22 Bestellbestätigung

Nach einer erfolgten Bestellung wird der Warenkorb geleert und eine neue Bestellung könnte aufgegeben werden. Der Lieferstatus kann in der Bestellübersicht eingesehen werden.

3.2.6 Inventur

Für die Umsetzung der Funktionalitäten zur Inventur wurde auf die Verwendung von Formularen gesetzt. Ist eine benutzende Person mit einem Administratorkonto angemeldet, kann diese auf die Funktionen der Inventur und Benutzerverwaltung zugreifen.

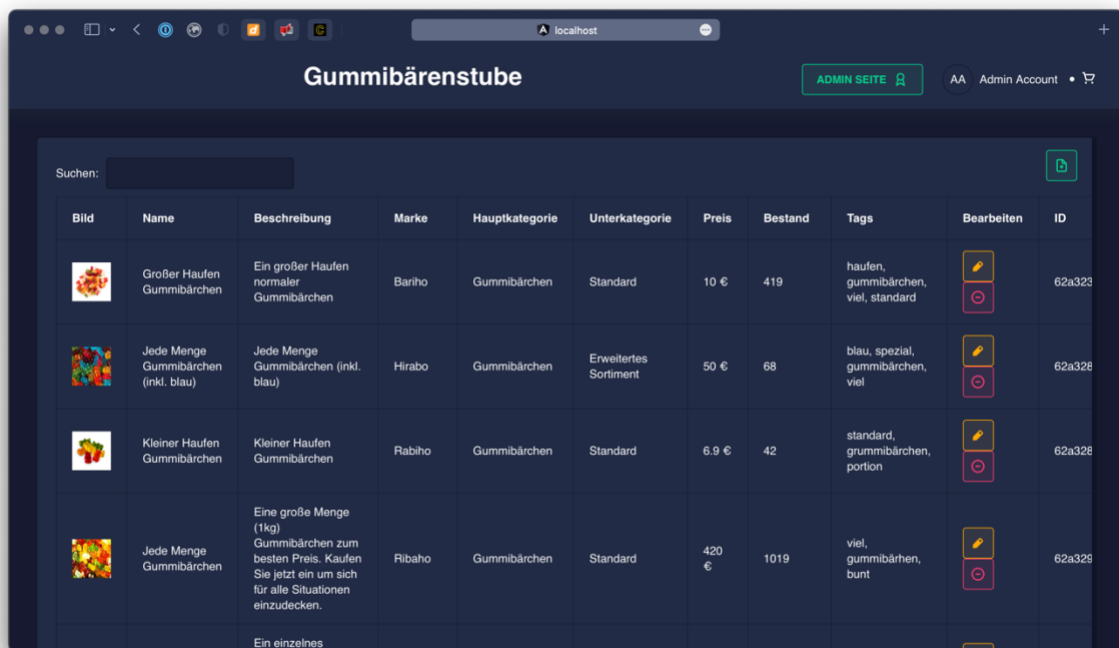















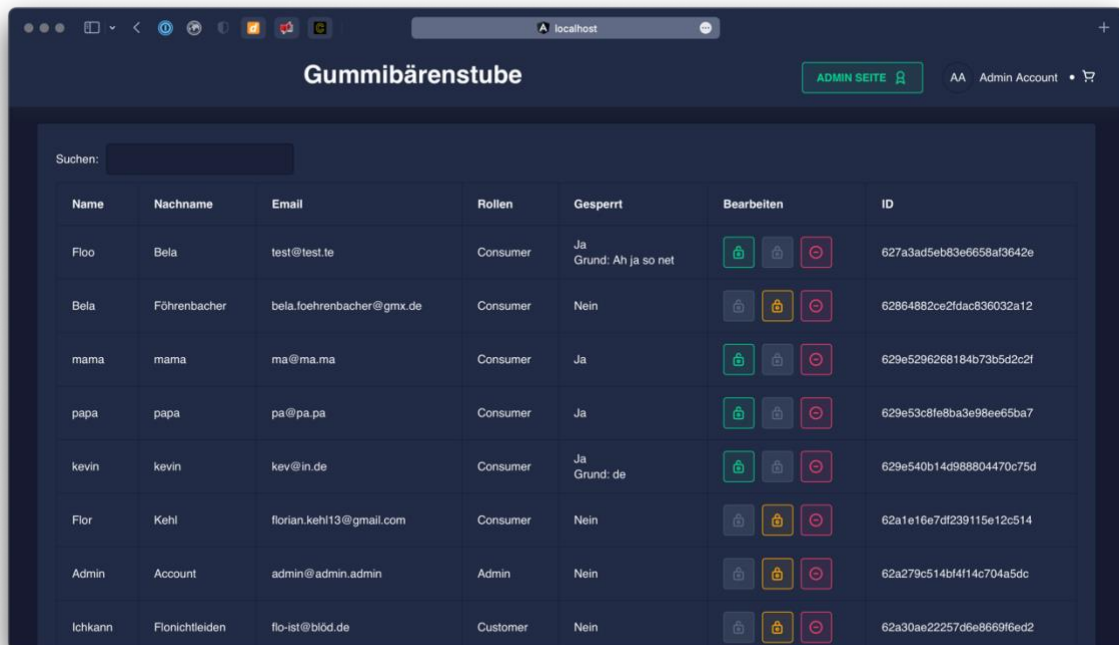
Bild	Name	Beschreibung	Marke	Hauptkategorie	Unterkategorie	Preis	Bestand	Tags	Bearbeiten	ID
	Großer Haufen Gummibärchen	Ein großer Haufen normaler Gummibärchen	Bariho	Gummibärchen	Standard	10 €	419	haufen, gummibärchen, viel, standard	 	62a323
	Jede Menge Gummibärchen (inkl. blau)	Jede Menge Gummibärchen (inkl. blau)	Hirabo	Gummibärchen	Erweitertes Sortiment	50 €	68	blau, spezial, gummibärchen, viel	 	62a328
	Kleiner Haufen Gummibärchen	Kleiner Haufen Gummibärchen	Rabiho	Gummibärchen	Standard	6.9 €	42	standard, grummibärchen, portion	 	62a328
	Jede Menge Gummibärchen	Eine große Menge (1kg) Gummibärchen zum besten Preis. Kaufen Sie jetzt ein um sich für alle Situationen einzudecken.	Ribaho	Gummibärchen	Standard	420 €	1019	viel, gummibärchen, bunt	 	62a329
		Ein einzelnes								

Abb. 23 Inventurfunktion mit Administratorkonto

In der Artikelverwaltung können bereits existierende Artikel angepasst werden. Der Preis lässt sich anpassen, die im Lager verfügbare Anzahl anpassen und Artikel können gelöscht oder hinzugefügt werden.

3.2.7 Benutzerverwaltung

Zusätzlich zu der Möglichkeit der Produktverwaltung, sollen auch Benutzende verwaltet werden können. Es soll die Möglichkeit geben Profile zu ent-/sperren oder zu löschen. Auch diese Funktion ist nur über die Administrationsseite aufzurufen.



The screenshot shows a web browser window with the title 'Gummibärenstube'. In the top right corner, there is a green button labeled 'ADMIN SEITE' and a user profile 'AA Admin Account'. Below the header is a search bar labeled 'Suchen:'. The main content is a table with the following columns: Name, Nachname, Email, Rollen, Gesperrt, Bearbeiten, and ID. The table contains eight rows of user data.

























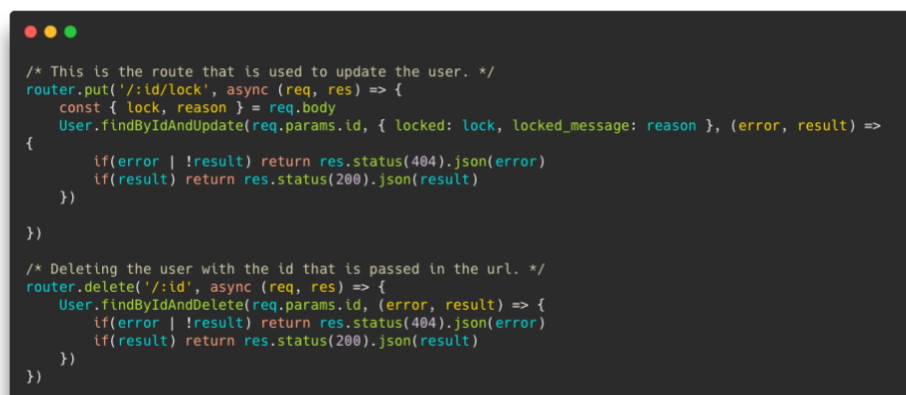
Name	Nachname	Email	Rollen	Gesperrt	Bearbeiten	ID
Floo	Bela	test@test.te	Consumer	Ja Grund: Ah ja so net	  	627a3ad5eb83e6658af3642e
Bela	Föhrenbacher	bela.foehrenbacher@gmx.de	Consumer	Nein	  	62864882ce2fdac836032a12
mama	mama	ma@ma.ma	Consumer	Ja	  	629e5296268184b73b5d2c2f
papa	papa	pa@pa.pa	Consumer	Ja	  	629e53c8fe8ba3e98ee65ba7
kevin	kevin	kev@in.de	Consumer	Ja Grund: de	  	629e540b14d98804470c75d
Flor	Kehl	florian.kehl13@gmail.com	Consumer	Nein	  	62a1e16e7df239115e12c514
Admin	Account	admin@admin.admin	Admin	Nein	  	62a279c514bf4114c704a5dc
Ich kann	Flonicht leiden	flo-ist@blöd.de	Customer	Nein	  	62a30ae22257d6e8669f6ed2

Abb. 24 Benutzerverwaltung

Abb. 25 zeigt einen kleinen Ausschnitt aus dem Backend wo die entsprechenden Datenbankabfragen zum Sperren oder Löschen von Konten gesendet werden.



```
/* This is the route that is used to update the user. */
router.put('/:id/lock', async (req, res) => {
  const { lock, reason } = req.body
  User.findByIdAndUpdate(req.params.id, { locked: lock, locked_message: reason }, (error, result) => {
    if(error || !result) return res.status(404).json(error)
    if(result) return res.status(200).json(result)
  })
})

/* Deleting the user with the id that is passed in the url. */
router.delete('/:id', async (req, res) => {
  User.findByIdAndDelete(req.params.id, (error, result) => {
    if(error || !result) return res.status(404).json(error)
    if(result) return res.status(200).json(result)
  })
})
```

Abb. 25 Backend Funktionen zum Löschen und Sperren von Konten

3.2.8 Statusänderung bei Bestellungen

Die letzte zusätzliche Funktion mit einem Administratorkonto ist die Möglichkeit den Status einer Bestellung zu ändern, um Benutzende über den Lieferstatus zu informieren. Hier könnten in Zukunft weitere Funktionen folgen, wie zum Beispiel das Bearbeiten der bestellten Artikel, bis zu einem gewissen Status.

The screenshot shows a web form titled 'Status Aktualisieren' on a dark blue background. It contains two main input sections: 'Derzeitiger Status' (Current Status) with a text box containing 'Lieferung fehlgeschlagen' (Delivery failed), and 'Neuer Status' (New Status) with a text box containing 'Erneuter Lieferversuch am 11.06.2022' (Retrial of delivery on 11.06.2022). Below these is a blue button labeled 'AKTUALISIEREN' (Update). At the bottom, there is a small text 't4c704a5dc' and a status indicator 'Bestellung erhalten' (Order received).

Abb. 26 Statusänderung von Bestellungen

4 Zusammenfassung und Ausblick

Das abschließende Kapitel der Vorliegenden Arbeit beinhaltet die wichtigsten Erkenntnisse aus der Umsetzung des Webshops, ein Fazit und einen Ausblick für die Zukunft dieses speziellen exemplarischen Webshops aber auch des online Handels allgemein.

4.1 Fazit & Erkenntnisse aus der Anwendung

Während der Umsetzung des Webshops wurden vor allem bestehende Grundlagen aus vorhergehenden Vorlesungsveranstaltungen verwendet und das Wissen darüber vertieft. Auch konnte für das Schreiben der Arbeit in den Vorlesungen erlerntes Wissen eingebracht werden, wie zum Beispiel die Grundlagen der Web Entwicklung und zu Datenbanken.

Darüber hinaus wurde festgestellt, dass für die durchschnittliche mittelständige Person, die einen Webshop für ihr eigenes Unternehmen benötigt, der Aufwand des Erstellens eines Webshops wie in dieser Arbeit beschrieben zu groß ist. Diese Personen sollten bevorzugt auf ein Baukastensystem zurückgreifen oder sich einen fertigen Webshop kaufen. Kompliziert und deshalb ausgelassen wurden daher auch die erweiterte Profilpflege und andere kleine Extras, die als erweiterte Funktionalität in Frage gekommen wären.

Zusammenfassend war die Entwicklung eines Webshops für Informatikstudenten im sechsten Semester eine gut umzusetzende Aufgabe, jedoch aufwändiger als zuvor erwartet. Diese Tatsache hat vor allem den Druck im Laufe der Zeit erhöht und gegen Ende des Projektes für erheblichen Mehraufwand gesorgt.

Die Erfahrung aus den Praxiseinsätzen in den Unternehmen zeigt, dass dies für viele Projekte der Fall ist und häufiger passiert als vielleicht angenommen.

4.2 Ausblick

Die Aussichten für einen Webshop wie dem in der vorliegenden Arbeit entwickelten, wären zum Beispiel, diesen auf einem Webserver zu hosten und über eine Domain aus dem Internet erreichbar zu machen. Dadurch wäre dieser 24 Stunden am Tag erreichbar und könnte realistisch zum Einsatz kommen.

Für einen tatsächlichen Einsatz wäre darüber hinaus ein Upgrade an der Datenbank vorzunehmen, um eine höhere Anzahl von Anfragen verarbeiten zu können, ohne an das Limit der kostenlos zur Verfügung gestellten Ressourcen zu stoßen. Darüber hinaus wäre in Erwägung zu ziehen, ob nicht eine relationale Datenbank in Frage käme, sobald die Datenstruktur für den Webshop und die enthaltenen Artikel und Kundenprofile final festgelegt wurde. Außerdem sollten zur Effizienzsteigerung die Produktbilder nicht als *Base64* gespeichert werden, sondern, wie bereits zuvor in dieser Arbeit erwähnt, zum Beispiel in einem *S3 Bucket* oder einer vergleichbaren Lösung.

Auch wären die im Laufe der Arbeit erwähnten, zusätzlichen Funktionen in Zukunft in den Webshop einzubringen, wie zum Beispiel detailliertere Filterung von Artikeln oder eine erweiterte Benutzerverwaltung, die auch das Bearbeiten von Benutzerkonten ermöglicht.

5 Quellenverzeichnis

- Heinemann, Gerrit (2016): Der neue Online-Handel, <https://books.google.com/books?hl=de&lr=&id=9tLJDQA AQBAJ&oi=fnd&pg=PR5&dq=online+handel+entwicklung&ots=Mf2yj3bQXp&sig=07a3vw8cjaPmg-Ci8XOBshzIIAY#v=onepage&q=online%20handel%20entwicklung&f=false> (11.05.2022, 09:43)
- Schicker, Edwin (2017): Datenbanken und SQL, <https://link.springer.com/content/pdf/10.1007/978-3-658-16129-3.pdf> (19.05.2022, 15:07)
- TechTerms.com (2013): NoSQL, <https://techterms.com/definition/nosql> (24.05.2022, 15:58)
- Akveo Inc. (2019): Nebular Homepage, <https://akveo.github.io/nebular/> (24.05.2022, 16:08)
- Schmidt, Leoni (2022): Eigene Website erstellen: 5 Tipps für den Start, <https://www.gruender.de/website/eigene-website-erstellen/> (30.05.2022, 09:28)
- IBM Cloud Education (2019): CAP Theorem, <https://www.ibm.com/cloud/learn/cap-theorem#toc-what-is-th-DXABpEgu> (30-05.2022)
- IBM Cloud Education (2020): Application Programming Interface (API), <https://www.ibm.com/cloud/learn/api> (10.06.2022, 16:43)
- TutorialKart (o. J.): Node.js Mongoose – Connect to MongoDB, <https://www.tutorialkart.com/nodejs/mongoose/connect-to-mongodb/> (10.06.2022, 16:34)

6 Glossar

- NoSQL: “NoSQL is a non-relational database that stores and accesses data using key-values. Instead of storing data in rows and columns like a traditional database, a NoSQL DBMS stores each item individually with a unique key. Additionally, a NoSQL database does not require a structured schema that defines each table and the related columns. This provides a much more flexible approach to storing data than a relational database.” [TechTerms.com (2013)]
- Nebular: “Nebular is a customizable Angular UI library that contains 40+ UI components, four visual themes, and Auth and Security modules. Recognized at the prestigious AngularConnect 2018, this Angular framework allows focusing on beautiful designs to adapt them to your brand. Nebular is free of charge and open-source.” [Akveo Inc. (2019)]
- API “An application programming interface, or API, enables companies to open up their applications’ data and functionality to external third-party developers, business partners, and internal departments within their companies. This allows services and products to communicate with each other and leverage each other’s data and functionality through a documented interface. Developers don’t need to know how an API is implemented; they simply use the interface to communicate with other products and services. API use has surged over the past decade, to the degree that many of the most popular web applications today would not be possible without APIs.” [IBM Cloud Education (2022)]