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](mailto: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](#_Toc104799251)

[Abbildungsverzeichnis IV](#_Toc104799252)

[Abkürzungsverzeichnis IV](#_Toc104799253)

[Ehrenwörtliche Erklärung V](#_Toc104799254)

[1 Einleitung 1](#_Toc104799255)

[1.1 Herausforderung 1](#_Toc104799256)

[1.2 Aufgabenstellung 1](#_Toc104799257)

[2 Theoretische Grundlagen 2](#_Toc104799258)

[2.1 Konzeptionierung 2](#_Toc104799259)

[2.2 Auswahl des Frameworks 3](#_Toc104799260)

[2.3 Auswahl einer Datenbank 4](#_Toc104799261)

[3 Implementation 5](#_Toc104799262)

[3.1 Designentwurf 5](#_Toc104799263)

[3.1.1 Grundsätzlicher Aufbau 5](#_Toc104799264)

[3.1.2 Detailansicht 6](#_Toc104799265)

[3.1.3 Nutzer Authentifizierung 7](#_Toc104799266)

[3.1.4 Admin Ansicht: Inventur 8](#_Toc104799267)

[3.1.5 Admin Ansicht: Benutzerverwaltung 9](#_Toc104799268)

[3.1.6 Bestellprozess & Einkaufswagen 11](#_Toc104799269)

[3.2 Praktische Implementierung 14](#_Toc104799270)

[3.2.1 Grundlegendes Design 14](#_Toc104799271)

[3.2.2 Detailansicht 15](#_Toc104799272)

[3.2.3 Nutzer Authentifizierung 15](#_Toc104799273)

[3.2.4 Inventur 16](#_Toc104799274)

[3.2.5 Benutzerverwaltung 17](#_Toc104799275)

[3.2.6 Bestellprozess & Einkaufswagen 18](#_Toc104799276)

[4 Zusammenfassung und Ausblick 19](#_Toc104799277)

[4.1 Erkenntnisse aus der Anwendung 19](#_Toc104799278)

[4.2 Fazit 19](#_Toc104799279)

[4.3 Ausblick 20](#_Toc104799280)

[5 Quellenverzeichnis 21](#_Toc104799281)

[6 Glossar 22](#_Toc104799282)

# Abbildungsverzeichnis

[Abb. 1 Skizze der Produktansicht 5](#_Toc104301038)

[Abb. 2 Skizze der Detailansich eines Produkts 6](#_Toc104301039)

[Abb. 3 Skizze der Login Seite 7](#_Toc104301040)

[Abb. 4 Skizze der Produktübersicht aus Sicht eines Adminkontos 8](#_Toc104301041)

[Abb. 5 Skizze der Admin Ansicht auf Benutzerkonten 9](#_Toc104301042)

[Abb. 6 Skizze der Bearbeitungsansicht eines Benutzerprofils 10](#_Toc104301043)

[Abb. 7 Skizze des Pop Ups zum Löschen oder Sperren eines Benutzerprofils 10](#_Toc104301044)

[Abb. 8 Skizze zur Übersicht der zu bestellenden Artikel 11](#_Toc104301045)

[Abb. 9 Skizze zur Adresseingabe im Bestellprozess 12](#_Toc104301046)

[Abb. 10 Skizze zur Auswahl der Bezahlmethode 13](#_Toc104301047)

# Abkürzungsverzeichnis

NoSQL **No SQL** / **N**ot **o**nly **SQL**

# 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.

Background pattern

Description automatically generated

Ort, Datum Unterschriften (Bela Föhrenbacher, Florian Kehl)

Stuttgart, 10.06.2022

# 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.

## 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.

## 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. Typische Funktionen wie Profilpflege und das Ändern von Passwörtern soll darüber hinaus auch möglich sein.

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.

# 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.

## 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.

## 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.

## 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)].

Darüber hinaus 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 (***C****onsitency*), Verfügbarkeit (***A****vailability*) und Partitionstoleranz (***P****artition Tolerance*) [vgl. IBM (2919)]. 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 (2919)], 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.

# 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.

## Designentwurf

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

### 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.

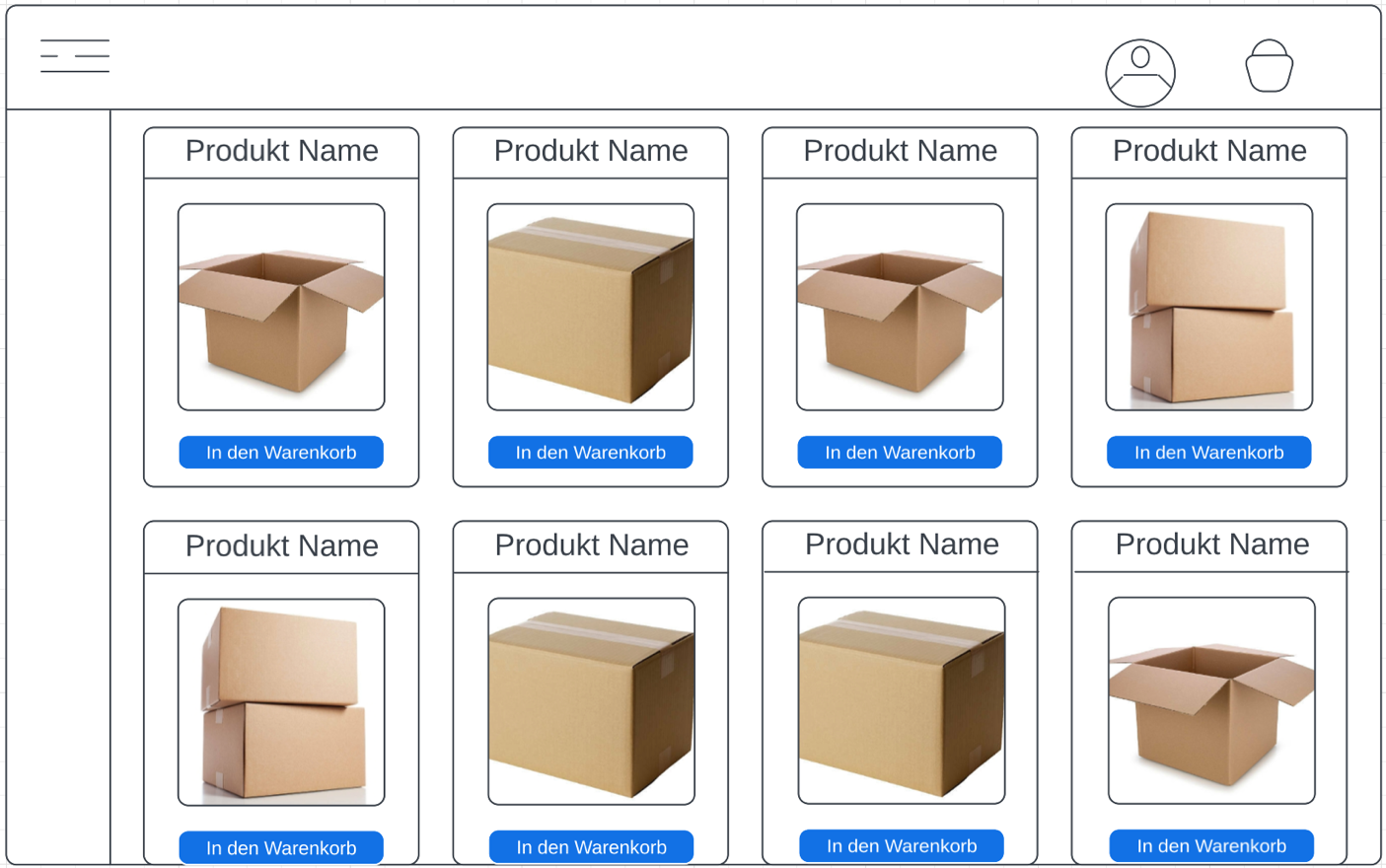


Abb. 1 Skizze der Produktansicht

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.

### Detailansicht

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

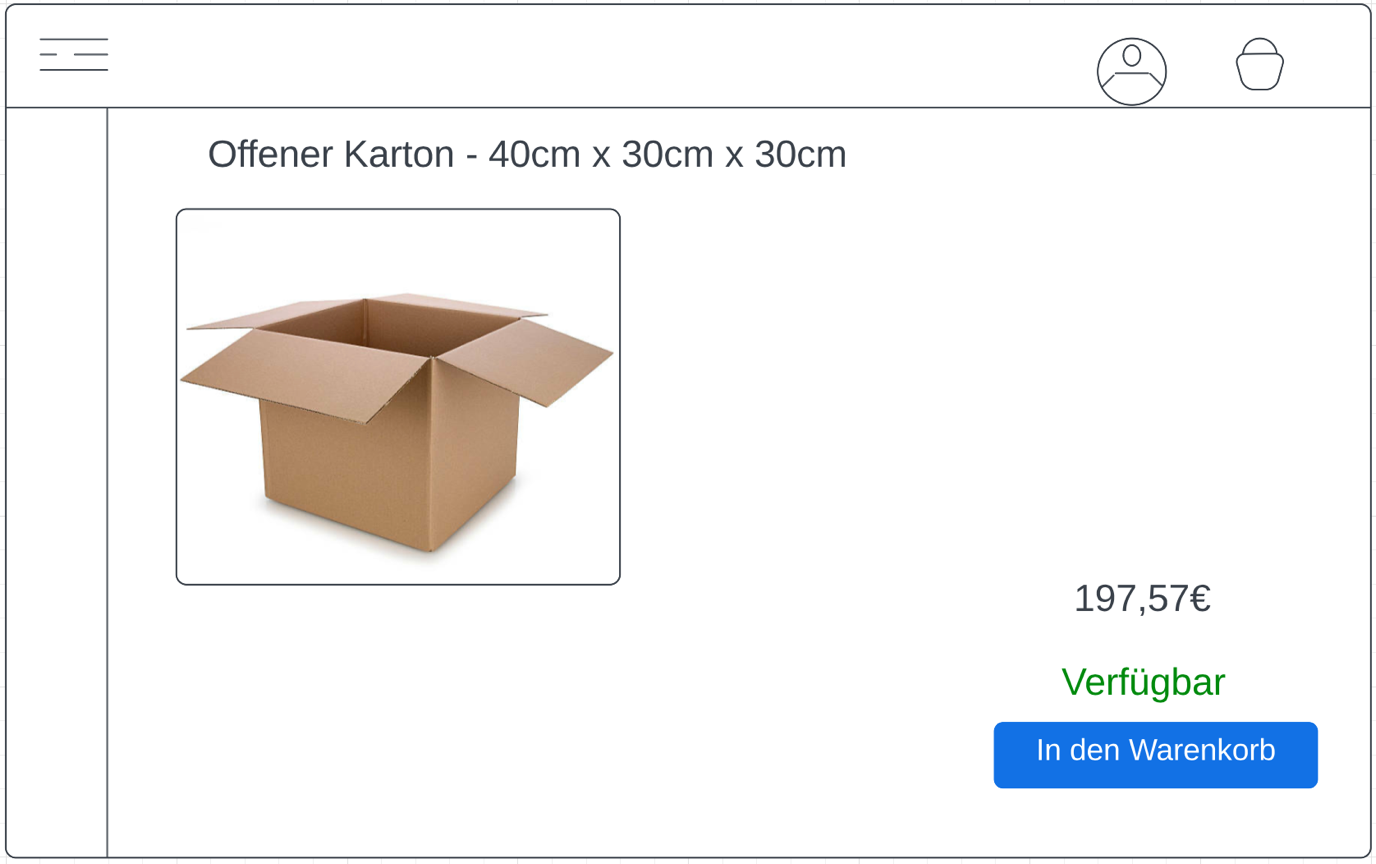


Abb. 2 Skizze der Detailansich eines Produkts

Hier sollen dann zum Beispiel genauere Verfügbarkeitsinformationen oder Lieferinformationen (Wie schnell kann Lieferung erfolgen? Lieferung mit LKW oder Transporter? Nur Abholung?) 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.

### 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.

Graphical user interface, application

Description automatically generated

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.

### 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.

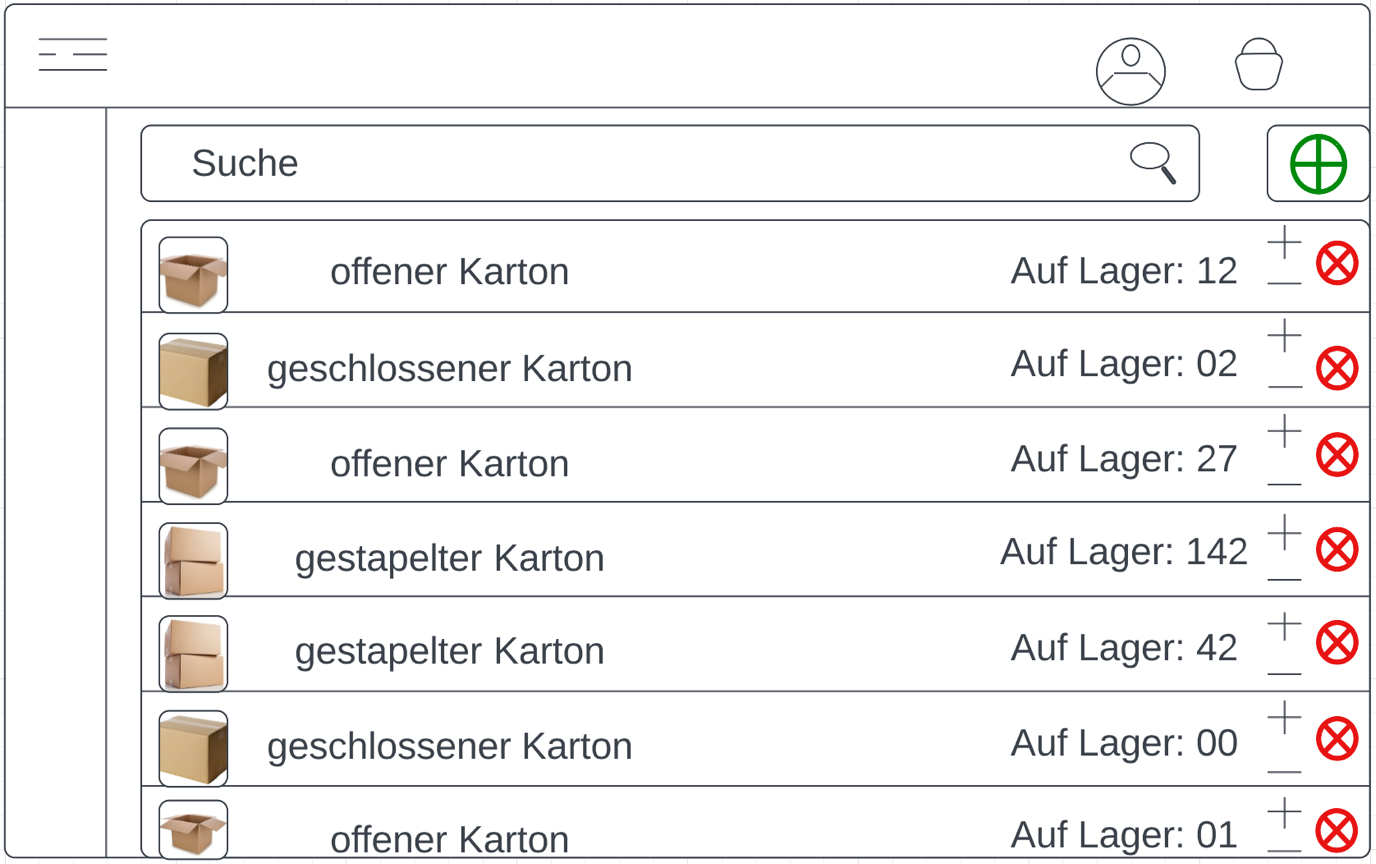


Abb. 4 Skizze der Produktübersicht aus Sicht eines Adminkontos

Abb. 3 zeigt eine grobe Skizze, wie diese Ansicht umzusetzen sein könnte. Die notwendigen Funktionen zur Inventur sind vorhanden.

### 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.

Graphical user interface, text, application, email

Description automatically generated

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.

Graphical user interface

Description automatically generated

Abb. 6 Skizze der Bearbeitungsansicht eines Benutzerprofils

Graphical user interface, text, application, email

Description automatically generated

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

### 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 besteht in einem weiteren Schritt die Möglichkeit den Warenkorb im Session Storage oder als Cookies zu speichern, damit auch nicht angemeldete Personen einen Einkaufswagen anlegen könnten.

Der Ablauf des Bestellprozesses soll 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.

A picture containing graphical user interface

Description automatically generated

Abb. 8 Skizze zur Übersicht der zu bestellenden Artikel

Hier soll wie in Abb. 4 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. 5 skizzierte Eingabe von Rechnungs- und Lieferadresse. Die Lieferadresse ist hierbei optional und muss nur angegeben werden, wenn sich diese von der Rechnungsadresse unterscheidet.

Text

Description automatically generated with medium confidence

Abb. 9 Skizze zur Adresseingabe im Bestellprozess

Nach der Eingabe von Liefer- und Rechnungsadresse (diese können bei angemeldeten Benutzenden automatisch ausgefüllt werden) erfolgt die Auswahl der Bezahlmethode, bevor eine finale Überprüfung der zu bestellenden Artikel durchgeführt wird.

Als Bezahlmethode soll, wie in Abb. 6 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.



Abb. 10 Skizze zur Auswahl der Bezahlmethode

Die Rechnungsstellung soll von einem node.js Modul übernommen werden, welches PDF-Dokumente erstellen kann. Die erstellte Rechnung wird dann per Mail zugesendet.

Table

Description automatically generated

Abb. 11 Beispielhaftes Design der Rechnung

## 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.

### Grundlegendes Design

Das grundlegende Design des Webshops wurde wie bereits zuvor geplant an den bekannten Designmustern großer Onlinehändler orientiert, mit einer Darstellung der Produkte in Kacheln und typischen Such- und Filterungsmöglichkeiten.

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, eine *Navbar* und als Hauptelement die Produktübersicht. Die Produkte werden in Kacheln 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.

### Detailansicht

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

### 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.

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.

### 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.

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.

### Benutzerverwaltung

…

### Bestellprozess & Einkaufswagen

…

# 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.

## 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.

## Fazit

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.

## 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.

# Quellenverzeichnis

Heinemann, Gerrit (2016): Der neue Online-Handel, <https://books.google.com/books?hl=de&lr=&id=9tLJDQAAQBAJ&oi=fnd&pg=PR5&dq=online+handel+entwicklung&ots=Mf2yj3bQXp&sig=07a3vw8cjaPmg-Ci8XOBshzIlAY#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)

# 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)]