DMSHop

Tom Medelaine

Inhalt

[Anforderungsanalyse 4](#_Toc90839965)

[Vorlage 4](#_Toc90839966)

[Artikel des Onlineshops 4](#_Toc90839967)

[Kategorien 4](#_Toc90839968)

[Suchleiste 5](#_Toc90839969)

[Warenkorb des Onlineshops 5](#_Toc90839970)

[Technische Anforderungen 5](#_Toc90839971)

[Datenbank 6](#_Toc90839972)

[PHP 6](#_Toc90839973)

[Umsetzung 7](#_Toc90839974)

[SQL 7](#_Toc90839975)

[Aufbau der Seite 9](#_Toc90839976)

[Navigation 9](#_Toc90839977)

[Seitencontent 9](#_Toc90839978)

[Produktextraktion 9](#_Toc90839979)

[Kategorisierung 10](#_Toc90839980)

[Searchbar 10](#_Toc90839981)

[Warenkorb 11](#_Toc90839982)

DM Shop

# Anforderungsanalyse

Es soll eine Website errichtet werden die folgenden Funktionen bietet:

* Onlineshopping
* Es sollen Artikel angezeigt werden die der Nutzer kaufen kann
* Es soll ein Warenkorb zur Verfügung gestellt werden der von dem Nutzer mit Gegenständen gefüllt werden kann
* Der Nutzer soll die Möglichkeit haben die Artikel der Website nach Kategorie zu filtern
* Eine Suchleiste wird bereit gestellt die dem Nutzer die Möglichkeit bietet nach dem Namen eines gewünschten Artikels zu suchen
* Es soll eine Übersichtsseite geben die dem Nutzer zeigt welche Gegenstände im Warenkorb vorhanden sind
* Die Übersichtsseite kann vom Nutzer bearbeitet werden und soll den Gesamtpreis des Einkaufs und dessen einzelne Artikel anzeigen

## Vorlage

Als Inspiration wurde von uns der schon existierende Online -Shop von der Drogeriekette DM gewählt. Dieser erfüllt alle unsere Anforderungen und soll uns als funktionierendes Beispiel dienen.

# Artikel des Onlineshops

Die Artikel die in dem von uns erstellten Onlineshop gelistet werden halten wir nah an der von uns gewählten DM Vorlage. Jeder Artikel wird kategorisiert, erhält einen Preis und eine Artikelbeschreibung. Zusätzlich bekommt jeder Artikel ein Bild das den beschriebenen Artikel bestmöglich abbildet. Es werden mehrere Artikel auf der Hauptseite angezeigt. Die Artikel sollen im Kartenformat, ähnlich unserer Vorlage, designt werden:

* klar voneinander abgegrenzt
* alle nötigen Informationen schnell ersichtlich
* für den Nutzer äußerst einfach zu bedienen

## Kategorien

Da unser Shop aus der Branche der Drogeriemärkte kommt werden auch dessen Artikel nächstmöglich an diesen ausgerichtet. Die Artikel werden deswegen in mehrere Kategorien eingeteilt, da wir uns an unsere Vorlage halten wird es sich hierbei um Kategorien handeln die auch die Drogeriekette von DM verwendet, dazu zählen:

Marken, Make-up, Pflege & Parfüm, Haare, Männerpflege, Gesundheit, Ernährung, Baby & Kind, Haushalt, Tier, Foto

Jeder Nutzer der Website soll durch diese Kategorisierung die Gegenstände oder Artikel die angezeigt werden filtern können um so schneller an die von Ihm oder Ihre gewünschten Produkte zu kommen.

## Suchleiste

Der online Shop soll über eine Suchfunktionalität verfügen. Diese Suchfunktionalität wird durch eine Searchbar bereitgestellt. In dieser Searchbar haben die Nutzer die Möglichkeit Buchstaben, Wörter und Zahlen einzugeben. Durch diese interaktive Eingabe hat der Nutzer eine zusätzliche Filtermöglichkeit. Die gesamten Artikel werden auf Grundlage Ihres Namens nach der Eingabe des Nutzers durchsucht und dann dementsprechend aufgelistet. Sollte die Suche des Nutzers zu keinem Treffer führen so werden auch keine Artikel angezeigt.

# Warenkorb des Onlineshops

Im Warenkorb des Onlineshops sollen alle Artikel abgelegt werden die der Nutzer kaufen möchte und dort hineingelegt hat. Dieser Warenkorb soll als Übersicht für den Nutzer dienen. In dieser Übersicht kann der Nutzer seinen Einkauf managen. Durch diesen Kontext soll der Warenkorb folgende Funktionalitäten bieten:

* Alle im Warenkorb befindlichen Produkte sollen aufgelistet werden
* Einzelne Artikel im Warenkorb sollen aus diesem entfernt werden können.
* Die Artikelbeschreibung, Artikelname, Bild und der Preis des Produktes sollen angezeigt werden
* Die Menge des jeweiligen Gegenstands soll angezeigt werden
* Die Anzahl der einzelnen Artikel im Warenkorb soll veränderbar sein (Stückzahl)
* Es soll der Gesamtpreis aller sich im Warenkorb befindlichen Artikel berechnen werden
* Der Gesamtpreis wird dem Nutzer angezeigt
* Der Bezahlvorgang soll eingeleitet werden können

# Technische Anforderungen

Als TechStack werden wir XAMPP verwenden da es unsere Anforderungen am besten erfüllt:

* Durch den in XAMPP enthaltenden Apache Webserver wird uns eine Plattform zu Verfügung gestellt auf der unsere Anwendung laufen kann
* Durch PHPMyAdmin wird uns eine Verwaltungssoftware für Datenbanken zu Verfügung gestellt auf der wir unser Backend aufbauen können
* Durch die Bereitstellung von PHP können wir unser Backend ansprechen und Operationen darauf durchführen
* Über HTML und CSS wird das Frontend unserer Anwendung entwickelt

## Datenbank

Die durch XAMPP bereitgestellt Datenbank ist eine MySQL Datenbank und soll als Grundlage für unser Backend dienen. Auf Ihr werden alle wichtigen Daten gespeichert und die vom Frontend gesendeten Befehle durchgeführt. Wir benötigen eine Tabelle die unsere Artikel enthält. Die Tabelle muss folgendes enthalten:

* **ID**, jeder Artikel benötigt eine eindeutige Nummer die über die sich ein Artikel identifizieren lässt
* **Name**, jeder Artikel soll einen Namen bekommen über den erkenntlich gemacht wird um was für einen Gegenstand es sich handelt
* **Kategorie**, jeder Artikel soll einer übergeordneten Kategorie zugeordnet werden können über die der Artikel gefunden werden kann
* **Beschreibung**, jeder Artikel soll über eine Beschreibung verfügen die das Produkt im Detail beschreibt
* **Preis**, jeder Artikel soll über einen Preis verfügen der den Wert des Produktes widerspiegelt
* **Bild**, jeder Artikel soll ein Produktbild beinhalten das hier in URL Form gespeichert wird
* **Code**, der Codeblock wird architektonisch benötigt um Datenoperationen am Artikel durchzuführen

Hier ein visuelles Beispiel der Tabelle:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Name | Code | Category | Description | Price | Image |
| 1 | Snack für Hunde | DogSnack01 | Tier | Hundefutter | 1.65 | http://hundebild... |
| 2 | Shampoo | Shampoo01 | Haare | Haar Waschmittel | 5,95 | http://shamppobild... |
| 3 | Zucker | Sugar01 | Ernaehrung | Biozucker für Genuss | 2,65 | http://zuckerbild... |

Da es sich um eine MySQL Datenbank handelt wird SQL verwendet um Operationen an der Datenbank vorzunehmen. Bevölkert wird die Datenbank vorlagennah an der DM Onlineseite. Wir werden hierzu Artikel von DM aussuchen und versuchen diese eigenständig in unsere Datenbank zu übertragen.

## PHP

Durch XAMPP können wir PHP verwenden um unser Frontend mit unserem Backend zu verbinden. PHP ermöglicht es uns mit seinen Klassen eine SQL-Verbindung zu unserer Datenbank aufzubauen. Durch diese Verbindung werden die Artikel aus der Datenbank an unser Frontend geschickt oder abgefragt um diese dort weiterzuverarbeiten und anzuzeigen.

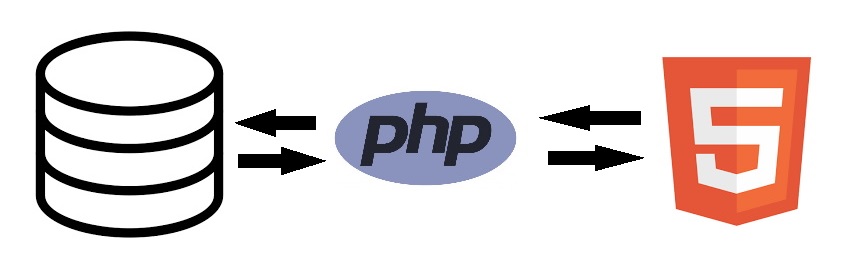
PHP ermöglicht es uns die SQL-Abfragen an unsere Datenbank zu schicken um die Artikel zu filtern die auf der Website angezeigt werden. Außerdem erlaubt es uns die Produkte die im Warenkorb gespeichert sind zu tracken.

Durch seine Möglichkeit Sessions zu erstellen wird uns PHP auch dabei helfen die Artikel des Warenkorbs ohne Datenbankabfragen zu verfolgen und dem User zuzuordnen.

PHP wird als Mittelpunkt der Anwendung genutzt:

Es verbindet die Formulare des HTML Contents auf dem Webserver mit den SQL-Abfragen an der Datenbank, liefert aber auch alle nötigen Daten vom Backend an die HTML Seiten.

# Umsetzung



## SQL

Die Datenbank die für unsere Anwendung von Nöten ist wird über einen SQL-Befehl erstellt. Um diesen Befehl ausführen zu können nutzen wir das von XAMPP bereitgestellt Datenbank Tool von phpMyAdmin. Hier erhalten wir eine grafische Nutzeroberfläche die es uns ermöglicht Datenbank bei GUI anzulegen oder aber auch SQL-Befehle zu nutzen. Mit folgendem SQL-Befehl wird die Datenbank erstellt:

SQL-Befehl:

**CREATE DATABASE** dmshop;

Wir haben mithilfe von SQL die Datenbank „dmshop“ erzeugt. Sie wird das Herzstück unseres Backend und enthält alle Tabellen die wir für unsere Anwendung benötigen. Da unsere Software Architektur so aufgebaut ist das wir eine Tabelle benötigen die die ganzen Artikel umfasst die im Onlineshop dargestellt werden sollen, müssen wir diese Tabelle auch erstellen. Hierfür verwenden wir wieder einen SQL-Befehl:

SQL-Befehl:

**CREATE TABLE IF NOT EXISTS**

**`products` (`id` int(10) NOT NULL AUTO\_INCREMENT,**

**`name` varchar(250) NOT NULL,**

**`code` varchar(100) NOT NULL,**

**`category` varchar(255) NOT NULL,**

**`description` varchar(255) NOT NULL,**

**`price` double(9,2) NOT NULL,**

**`image` varchar(250) NOT NULL,**

**PRIMARY KEY (`id`), UNIQUE KEY `code` (`code`)) ENGINE=InnoDB**

**DEFAULT CHARSET=latin1 AUTO\_INCREMENT=4 ;**

Mit diesem SQL-Befehl wurde die Tabelle „products“ erstellt. Sie verwaltet alle nötigen Daten um die Produkte oder Artikel unseres Onlineshops zu verwalten.

Da die Tabelle mit der Erzeugung noch keinem Eintrag hat, müssen wir diese erst noch hinzufügen, hierfür verwenden wir auch einen SQL-Befehl:

SQL-Befehl:

**INSERT INTO `products`**

**(`name`, `code`,`category`,`description`, `price`, `image`)**

**VALUES**

**( 'Titel des Artikels',**

**'Code für die Weiterverarbeitung',**

**'Kategorie',**

**'Beschreibungstext',**

**14.95,**

**'https://produktbild' )**

Mit diesem Befehl wird ein Eintrag der Produkttabelle hinzugefügt. Um einen weiteren Eintrag vorzunehmen muss der Befehl mit neuen Daten populiert und abgesendet werden.

Um die Daten von der Datenbank abzufragen wird ebenfalls ein SQL-Befehl verwendet. Dieser steht in Abhängigkeit der Anwendung und wird mit Hilfe von PHP erstellt. Um den Zusammenhang zwischen PHP und der Datenbank besser zu veranschaulichen wird im nächsten Abschnitt erklärt wie die Website aufgebaut ist und wie wir unsere Seite geplant haben.

# Aufbau der Seite

Die Onlineseite wird über die Index.php in 2 Hauptteile getrennt:

1. Navigation
2. Seitencontent

## Navigation

Die Navigation ist der obere Teil der Website. Hier findet der User alle relevanten Navigationselemente die Ihn über die komplette Seite leiten. Es befinden sich hier eine Auswahl der Kategorien die der Nutzer ansteuern kann, sowie eine Suchleiste in der es möglich ist nach dem Namen eines Artikels zu suchen. Auch befindet sich hier ein Button mit dem eine Navigation zum Warenkorb ermöglicht wird.

## Seitencontent

Der Seitencontent beinhaltet alles Wichtige im Seitencontext. Hier werden die Artikel aufgelistet die der Nutzer kategorisiert oder nach denen der Nutzer sucht. Es werden alle relevanten Informationen über die Artikel dargestellt und der Nutzer kann durch die Produkte browsen. Im Seitencontent wird auch der Warenkorb, über Aufruf der cart.php Seite, dargestellt, wenn der Nutzer sich dazu entscheidet seinem Einkauf mit dem Bezahlvorgang fortzuführen. Außerdem werden hier alle Formulare dargestellt die der User zur Eingabe benötigt, Lieferadresse, Rechnungsadresse etc.

# Produktextraktion

Um die Artikel aus der Datenbank zu extrahieren wird PHP verwendet. Hierfür wird als erstes eine Datenanbindung durch PHP vorgenommen. Es wird von uns eine dbcontext.php erstellt die diese Verbindung bereitstellt:

PHP-Code:

<?php

$con = mysqli\_connect("localhost","root","","dmshop");

    if (mysqli\_connect\_errno()){

        echo "Failed to connect to MySQL: " . mysqli\_connect\_error();

        die();

        }

?>

Die Datenbankverbindung wird über die mysqli\_connect Funktion hergestellt die in PHP konfiguriert ist. Sie nimmt als Parameter unsere notwendigen Datenbank Zugangsdaten und eröffnet eine Verbindung.

### Kategorisierung

Mit dieser Verbindung können wir nun unsere Daten extrahieren, hierzu werden die Nutzereingaben berücksichtigt. Unser Frontend bestimmt durch URL-Parameter welche Kategorie der Nutzer ausgewählt hat und das php sendet eine spezifische Abfrage in Form eines SQL-Befehls an unsere Datenbank. Hier ein Beispiel:

Wählt der Nutzer Gesundheit als Kategorie wird folgender SQL-Befehl über eine PHP-Switch Funktion an die Datenbank gesendet:

PHP-Code:

case 2:

*global* $sql;

$sql = "SELECT \* FROM products WHERE category='Ernaehrung';";

break;

Das Resultat dieser SQL-Abfrage wird wiederum in einem MySQL-Objekt gespeichert und von PHP verwendet um die entsprechenden Daten durch HTML Code anzuzeigen.

### Searchbar

Ähnlich handelt auch die Suchfunktion, wird eine Eingabe in der Suchfunktion abgesendet so wird in PHP auch eine SQL-Abfrage an die Datenbank gesendet, hierfür haben wir folgende Funktion erstellt:

PHP-Code:

function search($val){

if($val!= ""){

*global* $sql;

$sql = "SELECT \* FROM products WHERE name LIKE '%$val%';";

}

}

Diese Funktion nimmt den eingegebenen Text und führt einen LIKE Abgleich mit dem Namesfeld unser Produkttabelle durch, findet sich hier ein Treffer wird wieder ein MySQL-Objekt erstellt und der oder die Artikel werden angezeigt.

## Warenkorb

Der Warenkorb wird von PHP über eine SESSION Variable verwaltet. Hierfür ist es nötig das am Anfang eines Seitenaufrufs eine Session von PHP gestartet wird, in der alle relevanten Informationen Rund um die Artikel der der Nutzer in den Warenkorb gelegt hat, gespeichert werden.

PHP-Code:

session\_start();

$\_SESSION["shopping\_cart"] = $cartArray;

Es wird eine Session „shopping\_cart“ erstellt und in Ihr werden alle Artikel gespeichert die der Nutzer über das Hauptmenü seinem Einkaufswagen hinzufügt. Auf der Übersichtsseite cart.php wird diese Session dann aufgerufen und die Produkte darin dargestellt und können verändert werden.