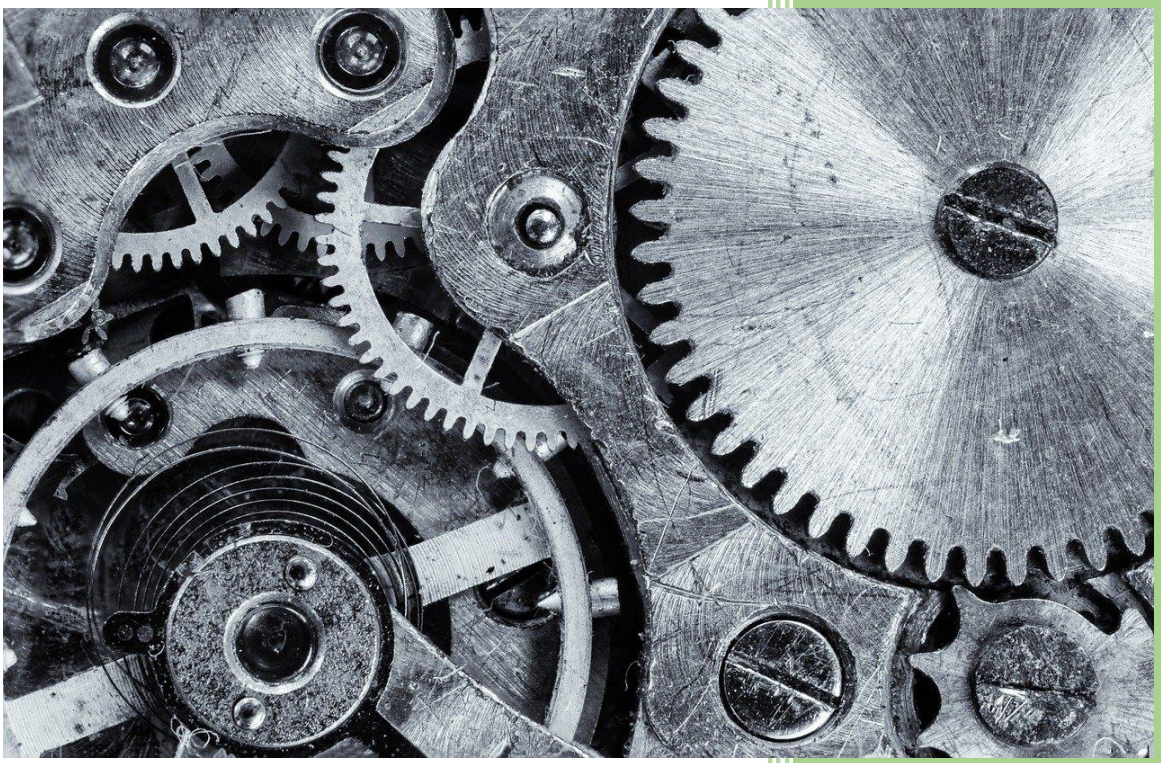
Dokumentation LB3



Jakov Ratkovic

Lauren Arekhi

Patrick Tomasi

Inhalt

Jakov Ratkovic
Lauren Arekhi
Patrick Tomasi

1 Allgemeines

Wir haben unsere Applikation den heutigen Anforderungen bezüglich Sicherheit angepasst. Das heisst es ist keine SQL-Injection möglich da wir mit Prepared Statements gearbeitet haben.

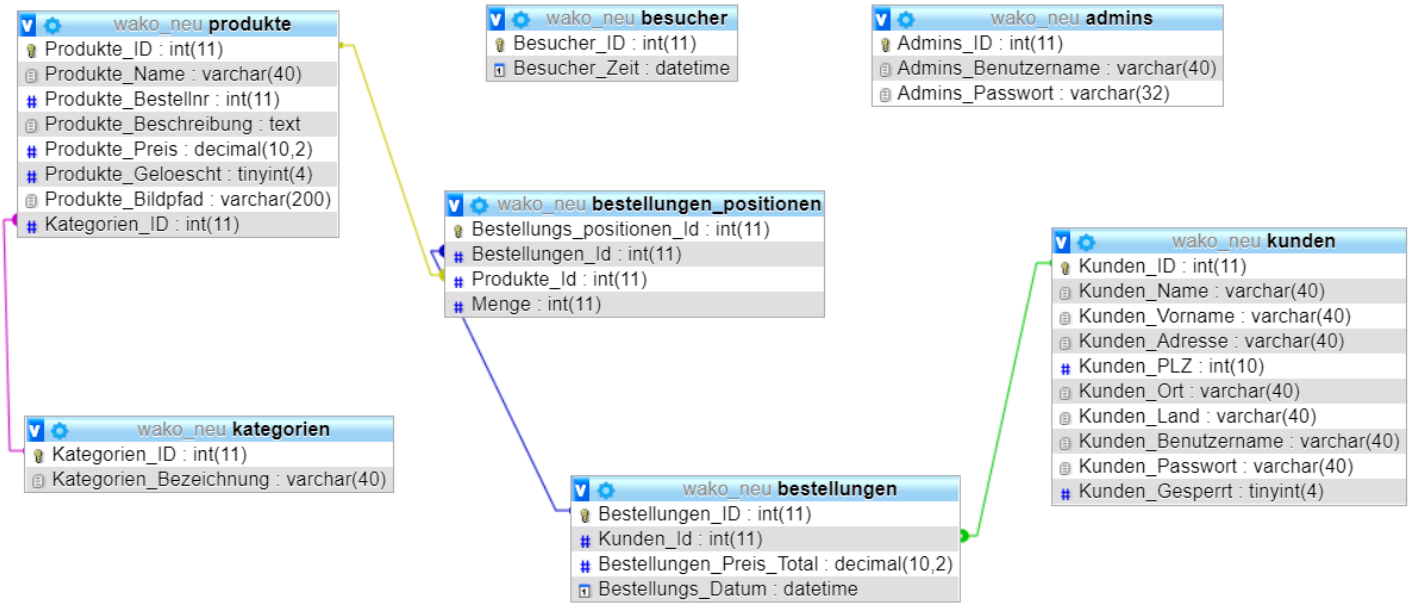
Zudem wurden alle Strings, die ausgegeben werden, UTF8 encoded damit es keine Probleme mit der Darstellung von Umlauten gibt.

Wir navigieren über sämtliche Seiten mithilfe von GET Parametern, die in der URL mitgesendet werden, und includen entsprechenden Content.

2 Datenbank

Die Datenbank haben wir komplett überarbeitet, da vieles fehlte oder fehlerhaft war.

Als erstes wurden alle nötigen Beziehungen mit den Tabellen hergestellt. Die Datenbank sieht nun wie folgt aus:



Jakov Ratkovic
Lauren Arekhi
Patrick Tomasi

Alle Tabellen wurden wie folgt überarbeitet:

2.1 Admins:

Admins_Id → AUTO_INCREMENT eingefügt
Admins_Benutzername → Erlaube Null auf Nein umgestellt
Admins_Passwort → Erlaube Null auf Nein umgestellt

2.2 Bestellungen:

Bestellungen_Id → AUTO_INCREMENT eingefügt

Kunden_ID → Neu hinzugefügt
Bestellungen_Preis_Total → Datentyp auf Decimal geändert
Bestellungs_Datum → Neu hinzugefügt

2.3 Besucher:

Besucher_ID → AUTO_INCREMENT eingefügt
Besucher_Zeit → Erlaube Null auf Nein und Standard current_timestamp() eingefügt

2.4 Kategorien:

Kategorien_ID → AUTO_INCREMENT eingefügt
Kategorien_Bezeichnung → Erlaube Null auf Nein umgestellt

2.5 Kunden:

Kunden_ID → AUTO_INCREMENT eingefügt
Kunden_Name → Erlaube Null auf Nein umgestellt
Kunden_Vorname → Erlaube Null auf Nein umgestellt
Kunden_Adresse → Erlaube Null auf Nein umgestellt
Kunden_PLZ → Erlaube Null auf Nein umgestellt
Kunden_Ort → Erlaube Null auf Nein umgestellt
Kunden_Land → Erlaube Null auf Nein umgestellt
Kunden_Benutzername → Erlaube Null auf Nein umgestellt
Kunden_Passwort → Erlaube Null auf Nein umgestellt
Kunden_Gesperrt → Datentyp von Int auf Tinyint umgestellt

2.6 Produkte:

Produkte_ID → AUTO_INCREMENT eingefügt
Produkte_Name → Erlaube Null auf Nein umgestellt
Produkte_Bestellnummer → Datentyp varchar auf Int umgestellt
Produkte_Preis → Datentyp von double auf Decimal und Erlaube Null auf Nein umgestellt
Produkte_Gelöscht → Datentyp von Int auf Tinyint und Erlaube Null auf Nein mit Standard 0 umgestellt

Die Tabelle Warenkoerbe wurde komplett entfernt und es wurde eine neue Tabelle bestellungen_positionen hinzugefügt.
Die Idee dahinter ist das jede Bestellung dann eine Kunden_ID hat um die Bestellung dem Kunden mit Namen, Adresse usw. zuordnen zu können. Und eine Produkte_ID hat damit das Produkt mit Preis usw. der Bestellung zugeordnet werden kann und eine Menge Spalte, um die Menge des Produktes zu speichern. Jede Bestellung hat somit eine Bestell_ID eine Kunden_ID und kann mehrerer bestelluneg_positionen haben.

3 Aufgabe 1 – Angebot Listenansicht

Die Aufgabe hier war nach einem Klick auf eine Kategorie alle Produkte dieser Kategorie aufzulisten. Dies haben wir so gemacht das man mit dem Klick auf eine Kategorie einen GET Parameter namens Kategorie mitsendet, dieser enthält die ID dieser Kategorie.

Danach werden alle Produkte mit dieser ID aus der Datenbank geholt, und über eine «While-Schlaufe» in eine Tabelle ausgegeben.

Man hat die Möglichkeiten bei einem Produkt auf die Detailansicht zu gehen oder direkt zur Warenkorb Ansicht zu gehen.

4 Aufgabe 2 – Angebote Detailansicht

Bei der Detailansicht war verlangt das nähere Informationen über das gewählte Produkt dargestellt werden. Das heisst es wird noch zusätzlich zu den bekannten Informationen der Beschreibungstext und ein entsprechendes Bild angezeigt.

Hier wurde auch mit einem GET-Parameter namens «Produkt» gearbeitet. Alle Angaben der Seite (inklusive des Bildes) werden vom GET-Parameter bestimmt. Dieser Parameter kann mit einer Prepared Statement dann je nach Anfrage die richtigen Daten aus der Datenbank holen.

Um von dieser Seite aus weiterzufahren stehen zwei mögliche Optionen zur Verfügung, entweder man geht per Icon zur Warenkorb Ansicht oder man kann zurück zur Hauptseite. Auf dem Weg zum Warenkorb wird die Produkte ID mitgegeben damit das richtige Produkt im Warenkorb angezeigt wird.

5 Aufgabe 3 – Angebote in den Warenkorb legen

Hier war verlangt das das man die Produkte, die man kaufen will in den Warenkorb legen kann.

Der Kunde kann in diesem Formular eingeben was für eine Menge des gewählten Produkts er haben will. Die Berechnung des Totalpreises wurde hier mit JavaScript und einem onchange EventListener umgesetzt, dies hat den Vorteil das die Seite nicht jedes Mal neu geladen werden muss um den Preis neu zu berechnen.

Es stehen einem zwei Optionen zur Verfügung, um weiterzufahren, entweder man legt das Produkt endgültig in den Warenkorb oder man kann zurück gehen.

Wenn das Produkt in den Warenkorb gelegt wird, wird es in die Session gespeichert damit der Kunde noch weiter einkaufen kann und am Schluss alle Produkte bezahlen kann.

Die Produkte werden im JSON Format in die Session gespeichert.

Beispiel. { "5": { "menge": 5, "produktPreis": 10}, "7": { "menge": 2, "produktPreis": 3.9}}

Die erste Zahl spiegelt die ProduktID wider und enthält ein Objekt das die Menge und der Preis der Bestellung enthält.

Zudem wird noch eine zweite Session geführt, die den totalen Betrag aller Bestellung enthält, damit der Kunde einen Überblick hat wie viel im Warenkorb ist.