**LB3 Dokumentation**

**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.

**Datenbank**

In der Datenbank haben wir die Beziehungen der Tabelle eingefügt da diese nicht vorhanden waren.

Zudem haben wir einige Datentypen der Felder geändert da diese nicht nicht korrekt waren.

Es wurde eine neue Tabelle Bestellungen\_positionen angelegt damit eine Bestellung zwischengespeichert werden kann bevor sie endgültig validiert wird.

**Aufgabe 1 – Angebot Listenansicht**

Die Aufgabe hier war nach eine 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.

**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.

Es stehen 2 Möglich Optionen zum weiterfahren zur Verfügung, entweder man geht zur Warenkorb Ansicht oder man kann zurück gehen.

**Aufgabe 3 – Angebote in den Warenkorb legen**

Hier war verlangt das das man 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 umgesetzt und einem onchange EventListener, dies hat den Vorteil das die Seite nicht jedes Mal neu geladen werden muss um den Preis neu zu berechnen.

Es stehen einem 2 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 wieder und enthält ein Objekt das menge und preis der Bestellung enthält.

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