

Documentation

Vorname	Wendelin
Name	Haller
Projekt	M151 LB

Inhaltsverzeichnis

- [Documentation](#)
 - [Inhaltsverzeichnis](#)
 - [Wichtige Dateine](#)
 - [Inbetriebname](#)
 - [Laravel Start](#)
 - [Created Modles](#)
 - [Insert PHPMyAdmin](#)
 - [Create Factorys](#)
 - [Views erstellen](#)
 - [Warenkorb erstellen](#)
 - [Erstellung des Adminbereichs](#)
 - [Unit Tests](#)
 - [Abname Tests](#)
 - [Warenkorb Test](#)
 - [Bestellungs Test](#)
 - [Issues](#)
 - [Aufgetauchte Fragen](#)
 - [Hilfsmittel](#)

Wichtige Dateine

- [Documentation PDF](#)
- [Daten.sql](#)

Inbetriebname

Docker starten und dann auf [localhost](#) gehen.

Als erstes muss der [SQL Code](#) auf der Datenbank ausgeführt werden. [PHPMyAdmin](#)

```
# Gast User:  
- email: gast@example.com  
- password: password  
  
# Admin User:  
- email: admin@example.com  
- password: klapp42stuhl
```

Laravel Start

Ich habe das Projekt an dem wir in der Schule gearbeitet haben kopiert und abgeändert.

Created Modles

Als erst habe ich alle Models und Migrations Dateien erstellt und den nötigen Code reingeschrieben.

Models:

- Artikel
- Bestell Position
- Bestellte Konfiguration
- Bestellung
- Inhalt
- Konfiguration
- Kunde
- User
- Warenkorb
- Warenkorb Artikel
- Warenkorb Artikel Inhalt

Insert PHPMyAdmin

[Daten.sql](#)

Diese Daten habe ich in der Datenbank gespeichert und natürlich auch die dazugehörigen Bilder im [public/img](#) Ordner.

Create Factorys

Ich habe für die meisten Models eine Factory erstellt um Test daten zu erstellen.

Views erstellen

Als erstes habe ich mit der Main Page angefangen danach habe ich die Artikel index gemacht wo man einen Überblick über alle Artikel hat. Danach habe ich mich um den Artikel show View gekümmert das die Artikel mit dessen Inhalten schön aufgezeigt werden. Zunächst habe ich das Erstellungs Formular für einen Kunden erstellt. Die nächste View die ich erstellt habe war die Kunde Show View und der Edit View. Für alle Views und Formulare habe ich noch die entsprechenden Controller erstellt und implementiert. Als nächstes habe ich die Darstellung und Speicherung des Warenkorbs erstellt. [siehe Warenkorb erstellen](#)

Warenkorb erstellen

Zuerst habe ich die drei Models erstellt. Und die dazugehörenden Migration und Controller implementiert.

Erstellung des Adminbereichs

Um einen Admin Bereich realisieren zu können habe ich beim User Model eine Spalte "is_admin" hinzugefügt um zwischen Kunde und Admin User unterscheiden zu können. Danach habe ich die CRUD Views und dazu die Controller implementiert. In den Views habe ich eine if Verzweigung verwendet

```
@if (Auth::user()->is_Admin())
```

um je nach dem ob er User Admin ist oder nicht die CRUD Formulare und Buttons anzuzeigen oder nicht.

Unit Tests

Ich habe 2 Tests erstellt für die ArtikelController Klasse und habe die Funktion "store" und "update" getestet.

Abname Tests

Zwei Abname Tests habe ich erstellt. Ein Warenkorb Test und ein Bestellung Test.

Warenkorb Test

	Testfall Nr.1
Testname:	Warenkorb
Absicht:	Artikel in den Warenkorb packen.
Eingabedaten:	Gast User Login-Daten
Soll-Ergebnis:	Auf der Home Seite wird der Warenkorb und die hinzugefügten Artikel angezeigt.

Schritt Nr.	Vorgehen	Erwartetes Ergebnis	Abweichungen
1	Als Gast User einloggen.	Die Home Seite wird angezeigt und man ist eingeloggt.	Keine
2	Auf den "Auswahl anzeigen." Knopf drücken.	Man kommt auf eine neue Seite und es werden alle Artikel angezeigt.	Keine
3	Auf den "Bestellen" Knopf eines Artikels drücken.	Die Seite des Artikels wird angezeigt.	Keine
4	Die einzelnen Inhalte auswählen und auf den "Bestellen" Knopf drücken.	Der Artikel mit den ausgewählten Inhalten wird in der Datenbank gespeichert und die Home Seite wird angezeigt mit dem Warenkorb und seinen Artikel und deren Inhalten.	Keine

Bestellungs Test

	Testfall Nr.2
Testname:	Bestellung
Absicht:	Was im Warenkorb ist zu bestellen.
Eingabedaten:	Angemessene Kunden Daten.
Soll-Ergebnis:	Der Warenkorb ist leer und die Bestellung ist in der Datenbank gespeichert.

Schritt Nr.	Vorgehen	Erwartetes Ergebnis	Abweichungen
1	Testfall Nr.1 durchführen.	Auf der Home Seite wird der Warenkorb und die hinzugefügten Artikel angezeigt.	Keine
2	Auf den "Check out" Knopf drücken.	Es wird ein Kunden-Formular angezeigt.	Keine
3	Das Kunden-Formular ausfüllen und auf den "Weiter" Knopf drücken.	Zur Überprüfung werden die Kunden-Daten und der Warenkorb angezeigt. Es wird auch ein Eingabefeld für Bemerkungen angezeigt.	Keine
4	Eine Bemerkung hinzufügen und auf den "Bestellung abschliessen" Knopf drücken.	Es wird eine Dankens-Seite angezeigt. Die Bestellung wurde in der Datenbank gespeichert. Der Warenkorb wurde gelöscht.	Keine

Issues

Was habe ich im Verlauf des Projektes für Probleme.

- Ich habe zwei Spalten bei den Models vergessen zu erstellen was dann zu Fehler geführt hat.

Aufgetauchte Fragen

Im Verlauf des Projekts sind ein Paar Fragen aufgetaucht.

- Wenn als erstes die Produkt_Position erstellt werden muss, dies aber ein Foreign Key von Bestellung braucht / wie kann man temporär zwischen verschiedenen Views Daten speichern Kunde Id ...
 - Warenkorb erstellen
- Wie erstellt man einen Admin-User ?
 - User eine Spalte is_admin hinzufügen.
- Wie soll man das mit dem Kunden Infos und zurück machen Edit-View oder Create view mit ausgefülltem Formular.
 - edit View !!
- Show Kunde View anders lösen nicht mit show und id in Url
 - show und id in URL und überprüfen ob der richtige user eingeloggt ist.

Hilfsmittel

- [Laravel Dokumentation](#)
- [Stackoverflow](#)
- [nelkasovic/docker-laravel-playground](#)
- Lehrer der mir Frage beantwortet hat.