

# Dokumentation och Användare manual till Shoppinglist

Jag valde att bygga applikationen objektorienterad vilket innebär att de grundläggande sett bygger på klasser och *Model*, *View* och *Controller* (MVC).

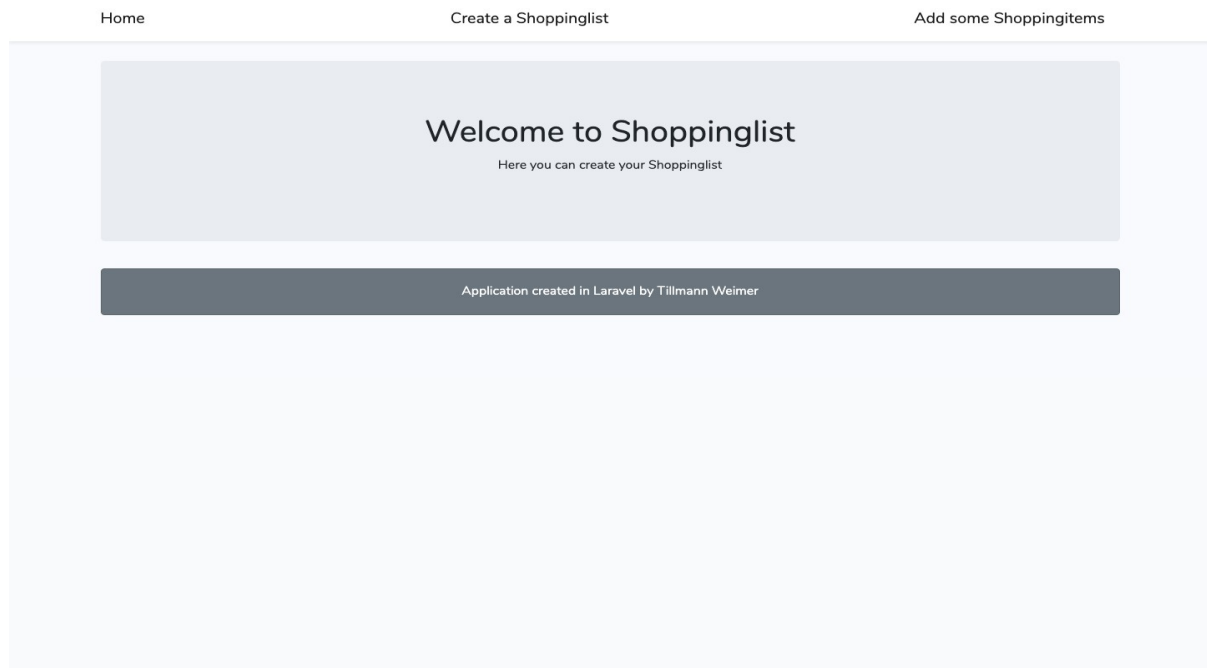
## Lite grundläggande om MVC

- *M. (Model)* kommunicerar med databasen vilket betyder att det är här man definerar vilka kolumner ska användas vid insert, updating eller deleting av data. Det är här man skapar relationer mellan entiter som är viktiga vid hämtning och filtrering av data.
- *V. (View)* tar hand om blade filerna. Det är det som visas som formulär. Dessa filer använder *blade syntax* men är skrivna i HTML.
- *C. (Controller)* innehåller de metoderna som behövs för att kommunicera både med Model och View. Varje entitet har sin egen Controller som extender (ärva) från klassen Controller.

## Shoppinglist without Authentication

Eftersom sidan inte är live kommer denna korta presentation:

Jag gjorde denna applikation objektorienterad med Laravel 7 som MVC ramverk och använde Bootstrap för layout.



*The landingpage*

Denna Applikation bygger på två entiteter:

- Shoppinglists
- Shoppingitems

Dessa entiteter är kopplat ihop genom relationer i modellen. En Shoppinglist kan ha många Shoppingitems (*one to many*).

I Shoppinglist model finns en metod (*shoppingitems*) som sköter det genom *hasMany* relationen. I Shoppingitem model finns den spegelvända relationen *belongsTo*.

När man skapar ett Shoppingitem kommer man till följande sida. Den visar Shoppinglists under formuläret som är klickbar.

The screenshot shows a web application interface. At the top, there are three navigation links: 'Home', 'Create a Shoppinglist', and 'Add some Shoppingitems'. The main content area is divided into two sections. The top section, titled 'Add some Shopping items', contains a form with the following fields: 'Category' (a dropdown menu with 'Select a Category' as the placeholder), 'Shoppinglist' (a dropdown menu with 'Select a Shoppinglist' as the placeholder), 'Name' (a text input field with 'Name of your Shoppingitem' as the placeholder), and 'Quantity' (a text input field with 'Number of Shopping Items' as the placeholder). Below these fields is a blue button labeled 'Save your Shopping Item'. The bottom section, titled 'Your Shoppinglists', displays a list of two items. Item 1 is 'Dagens lista' with a checkbox next to it. Item 2 is 'Fredagslistan' with a checkbox next to it. Below the list are two buttons: a blue button labeled 'Add some Shoppingitems' and a red button labeled 'Delete selected Shoppinglist'. At the bottom of the page, there is a footer that reads 'Application created in Laravel by Tillmann Weimer'.

*View for creating Shoppingitems*

När man klickar på ett Shoppingitem kommer man till följande sida (*show.blade.php*)

The screenshot shows a web application interface. At the top, there are three navigation links: 'Home', 'Create a Shoppinglist', and 'Add some Shoppingitems'. The main content area is divided into two sections. The top section, titled 'Fredagslistan', displays a list of two items. Item 1 is 'Article: Chips' with a checkbox next to it, and its details are 'Category: snack' and 'Quantity: 2'. Below this item is a blue button labeled 'Edit Shoppingitem'. Item 2 is 'Article: Beer' with a checkbox next to it, and its details are 'Category: other' and 'Quantity: 5'. Below this item is a blue button labeled 'Edit Shoppingitem'. Below the list is a timestamp that reads 'list created at: 2020-06-12 10:56:10'. At the bottom of the page, there are three buttons: a blue button labeled 'Back', a blue button labeled 'Add some Shoppingitems', and a red button labeled 'Delete selected Item'. At the bottom of the page, there is a footer that reads 'Application created in Laravel by Tillmann Weimer'.

Här kan man via en checkbox ta bort Shoppingitems eller redigera ett befintligt item.

Jag har skapat översättningssträngar (*i18n*) som skulle kunna möjliggöra översättning till andra språk om detta önskas vid senare vidareutveckling.

I Laravel finns det goda möjligheter att hantera och utveckla en databas via migrations vilket betyder att alla ändringar man gör i databasen görs i migrations filer som kan nås via *app/database* mappen. Det ger gode möjligheter om man utvecklar en applikation i ett team.

## Manual för uppsättning i ett lokalt miljö

Du är välkommen att kлона ner applikationen från mitt git repo [här](#).

Efter du har klonat ner git repot ska du skapa en lokal databas i till exempel *PHP myadmin*.

Det är bra att köra *composer install* kommandon i terminalen först. Det behövs för att kunna starta servern med *php artisan serve* kommandon i terminalen.

I mitt projekt finns det en fil med namnet *.env.exempel*. I denna fil skapar men förbindelse med databasen i att ange *database name*, *username* och *password* under *DB\_CONNECTION=mysql* (rad 9). Det är viktigt att du omdöpar filen till *“.env”* sen.

På rad 3 i filen finns det *“APP\_KEY”* som är tom. Då ska du skapa en key med kommandon *php artisan key:generate*. Detta skapar en nyckel som behövs för att kunna köra applikationen lokalt.

Det finns dessutom en fil med namn *database.php* där du också ska ange *database name*, *username* och *password* i sektionen *mysql* (rad 46). För att färdigställa databasen ska du köra *php artisan migrate:install* kommandon i terminalen. Med detta skapas en migration table. Det är en slags logfil för alla migrationer.

När detta är gjort kan du köra *php artisan migrate:status* i terminalen som visar genomförda eller icke genomförda migrationer. Den följande bilden visar ett exempel på icke genomförda migrationer:

```

/Applications/AMPPS/www/shoppinglist.test — php • php artisan serve
+-----+-----+-----+
| Ran? | Migration | Batch |
+-----+-----+-----+
| No   | 2014_10_12_000000_create_users_table |      |
| No   | 2014_10_12_100000_create_password_resets_table |      |
| No   | 2019_08_19_000000_create_failed_jobs_table |      |
| No   | 2020_06_08_085432_create_shoppinglists_table |      |
| No   | 2020_06_08_085457_create_shoppingitems_table |      |
+-----+-----+-----+
Julias-MacBook-Air-2:shoppinglist.test juliake$
```

Nu kan du köra *php artisan migrate* för att genomföra migrationen. I denna process blir själva databasen skapat. Om man kör *php artisan migrate:status* i terminalen nu får man följande:

```
Applications/AMPPS/www/shoppinglist — php • php artisan serve
Applications/AMPPS/www/shoppinglist — bash

+-----+-----+-----+
| Ran? | Migration | Batch |
+-----+-----+-----+
| Yes  | 2014_10_12_000000_create_users_table | 1 |
| Yes  | 2014_10_12_100000_create_password_resets_table | 1 |
| Yes  | 2019_08_19_000000_create_failed_jobs_table | 1 |
| Yes  | 2020_06_08_085432_create_shoppinglists_table | 1 |
| Yes  | 2020_06_08_085457_create_shoppingitems_table | 1 |
+-----+-----+-----+
Julias-MacBook-Air-2:shoppinglist juliake$
```

Efter du har skrivit *php artisan serve* i terminalen (om du inte har gjort innan) skulle du gärna komma till startsidan om du går in på localhost:8000.