

# Dokumentation och Användare manual till Shoppinglist with inloggning

För att ta hänsyn till säkerhet skapade jag ett inloggningssystem (*authentication*).

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

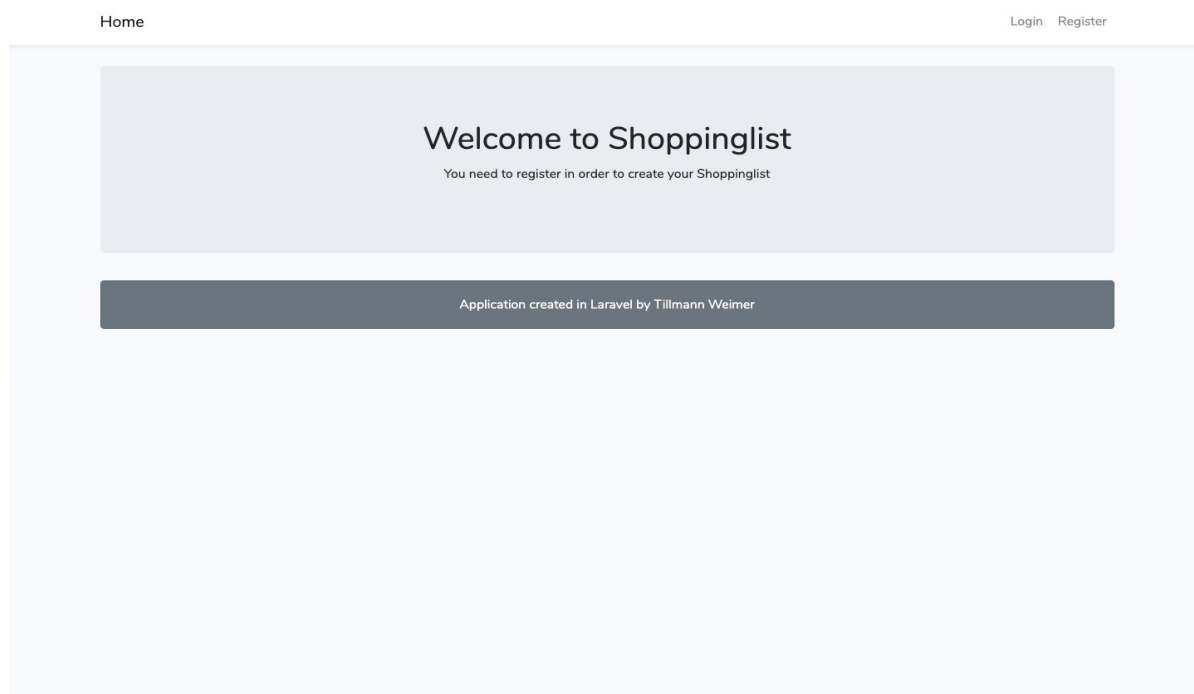
## Lite grundläggande om MVC

- *M. (Model)* kommunicerar med databasen vilket betyder att det är här man definerar vilka kolumner ska användas vid inserting, 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 filerne. Det är det som visas som formulär. Dessa filer använda *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 extendar (ärva) från klassen Controller.

## Shoppinglist with 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 landingspage*

Denna Appliktion bygger på tre entiteter:

- User

- Shoppinglists
- Shoppingitems

När man kommer till landningssidan behöver man registrera en användare.

The screenshot shows a web application interface. At the top, there is a navigation bar with 'Home' on the left and 'Login Register' on the right. The main content area is a light blue rectangle. Inside this area is a white box with a light gray header labeled 'Register'. Below the header, there are four input fields: 'Name', 'E-Mail Address', 'Password', and 'Confirm Password'. Each field has a corresponding label to its left. Below the 'Confirm Password' field is a blue button with the text 'Register' in white.

*registreringssidan*

Efter man har registrerat en användare och loggat in kommer man till följande sida

The screenshot shows the user dashboard after login. At the top, there is a navigation bar with 'Home' on the left and 'Test Testsson' with a dropdown arrow on the right. The main content area is a light blue rectangle. Inside this area is a white box with a light gray header labeled 'Welcome Test Testsson'. Below the header, there is a message: 'You have no Shoppinglists yet. You can create your first one [here](#)'. At the bottom of the page, there is a dark gray footer bar with the text 'Application created in Laravel by Tillmann Weimer'.

Här ska man skapa sin första Shoppinglist för att sedan kunna lägga till Shoppingitems. Man kan även klicka på Använderens namn uppe till höga (*se bilden*).

[Home](#)[Test Testsson](#)

Create a Shoppinglist  
Add a Shoppingitem to a list  
Logout

Create a Shoppinglist

Title

The title of your Shoppinglist

Save your Shopping list

No Shoppinglists

Application created in Laravel by Tillmann Weimer

*Formuläret för create Shoppinglist ('/shoppinglists/create.blade.php')*

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

[Home](#)[Create a Shoppinglist](#)[Add some Shoppingitems](#)

Add some Shopping items

Category

Select a Category

Shoppinglist

Select a Shoppinglist

Name

Name of your Shoppingitem

Quantity

Number of Shopping Items

Save your Shopping item

Your Shoppinglists

1.

☐ Dagens lista

2.

☐ Fredagslistan

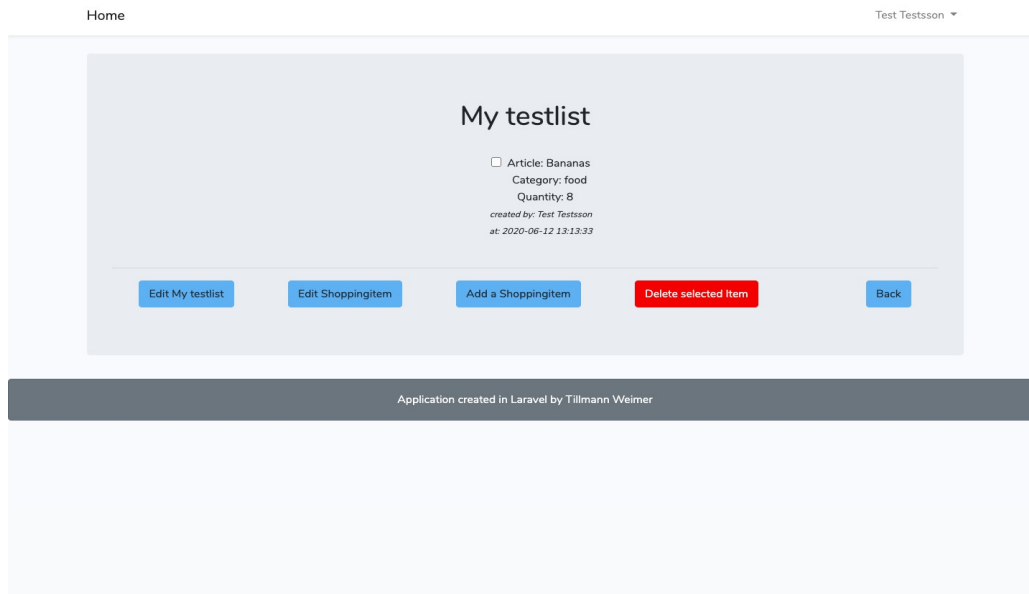
Add some ShoppingitemsDelete selected Shoppinglist

Application created in Laravel by Tillmann Weimer

*View for creating Shoppingitems*

Den visar endast Shoppinglists som är skapade av den inloggade användare.

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



Den visar endast en Shoppinglist med Shoppingitem som är skapat av den inloggade användare. 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
+-----+-----+-----+
| 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.

Tillmann Weimer  
0708280833  
tillmann1970@gmail.com