

Bilia

Hyr en bil hos oss

1. Vi behöver WAMP64 (som använder PHP version 7.2.33 och uppåt), BASH GIT, VISUAL STUDIO COD, EN DATA (JAG HAR WINDOWS).
2. I GIT hamnar jag på www mapen : `cd c/wamp64/www`
3. Det första, skaffar jag en Laravel projekt med Bilia namn som jag har valt :
`c/wamp64/www composer create-project laravel/laravel Bilia --prefer-dist`
4. Nu öppnar jag mappen (Bilia) med VISUAL STUDIO COD och skaffar modeller. Det första behöver jag skaffa färg och märke för bilar som vill hyra ut via bash git i Bilia mappen. Sedan skaffar modellen till bilar och kunder.
`>www >Bilia>php artisan make:model Color -m`
`>www >Bilia>php artisan make:model Make -m`
`>www >Bilia>php artisan make:model Car -m`
`>www >Bilia>php artisan make:model Customer -m`
5. Nu har vi modeller och migrationer i VISUAL STUDIO COD. Nu ska skaffa tabeller för våra modeller.
För färgen behöver jag bara en sträng (titel) som visar färger.
För märke behöver jag också bara en sträng (titel) som visar märke.
För bilar behöver jag tabeller som står på instruktören och för kunderna också.
6. På `localhost/phpmyadmin` öppnar jag `MYSQL` och skaffar en `database` med `Bilia` namn.
7. I `database>seeders` matar in jag mina database angående färger och maker. Nu man tillåter bara välja färger och maker som jag skaffade innan.

8. Nu ska jag göra migrate att skicka tabeller till database:

```
php artisan migrate
```

eftersom fick jag hinder ska ge jag ett par koder:

```
php artisan cache:clear
```

```
php artisan config:clear
```

```
php artisan config:cache
```

```
php artisan route:cache
```

```
php artisan migrate:fresh
```

9. Nu har jag mina tabeller på database i MYSQL.

10. Nu ska jag skaffa Controller (det går att köra vidare utan controller men är det bättre att ha den). På

bash git gör jag så:

```
Php artisan make:controller (exemplevis ) CustomerController
```

Nu har [CustomerController](#) jag på [App\Http\Controllers](#)

Och

```
CarController
```

```
checkController
```

```
colorController
```

```
HistoryController
```

11. Nu skaffar jag HTML filler `index.blade.php` , `main.blade.php` och `nav.blade.php` .

Jag kommer att använda **bootstrap** för att göra snyggare sidor.

12. Med VISUAL STUDIO COD på `\resources\views` skaffar jag:

```
index.blade.php
```

```
main.blade.php
```

```
nav.blade.php
```

13. I [nav.blade.php](#) kommer jag att låna koder av **bootstrap** och kopplar den till [main.blade.php](#)

14. I [main.blade.php](#) kommer jag att låna koder av **bootstrap** också.

15. På `App>Http>Controllers`

Här kommer jag skriva koder att styra och validitera.

16. I database>migrations har jag mina tabeller för colors, makes, cars, customers och histories.

17. På view> skaffar mappar och filer :

Mapp: [car](#) och filer: [add.blade.php](#), [edit.blade.php](#) och [index.blade.php](#)

Mapp: [check](#) och filer: [checkin.blade.php](#) och [checkout.blade.php](#)

Mapp: [customer](#) och filer: [add.blade.php](#), [edit.blade.php](#) och [index.blade.php](#)

Mapp: [history](#) och fil: [index.blade.php](#)

18. På routes>web ska jag ge router att ha metoden i :

19. Om nån ta bor en bil från tabellen när bilen är uthyrd då blir det katastroff. Jag ska fixa den. Gör så:

I resources>views>Car>index.blade.php:

```
@if($car->check_out_by == null)
    <a href="{{url('car/remove/'. $car->id)}}" class="btn btn-danger btn-sm">Remove</a>
@endif
```

Nu man har inge knapp att ta bort en upptagen bil. Men det är inte omöjligt. Man kan koppera länken från andra bil och byra nyckeln och tabort bilen. Så jag fixar den i Controller också. Desutom när man ta bort en lidig bil ska tas bort från history också:

I app>Http>Controller>CarController:

```
public function remove($id){
    $car = Car::find($id);

    if ($car->check_out_by == null) {
        $histories = History::where('car_id', $id)->get();
        foreach($histories as $history)
            $history->delete();

        $car->delete();
    }

    return redirect('cars');
```

20. När man vill ta bor en bil ska få varning alert. Jag kommer att använda fixa det i remove.

I resources>views>Car>index.blade.php:

```
<a href="{{ url('car/edit/'. $car->id)}}" class="btn btn-primary btn-sm">Edit</a>
@if($car->check_out_by == null)
```

```

<a
href="{{url('car/remove/'. $car->id)}}"
class="btn btn-danger btn-sm"
onclick="return confirm('Are you sure to delet the car?')"
>Remove</a>
@endif

```

21. Det är bättre att skaffa ett system för meddelande dessutom använder jag Bootstrap.
I View>main.blade.php jag kommer att bli redo för meddelande:

```
@if(Session::has('message'))
```

```

<div class="alert alert-warning alert-dismissible fade show" role="alert">

    <button type="button" class="close" data-dismiss="alert" aria-label="Close">

    <span aria-hidden="true">&times;</span>

    </button>

    {{Session::get('message')}}

</div>

```

```
@endif
```

22. Om man skriver fel då behöves det att man varnas om validationen. Jag kommer att använda igen Bootstrap att varna:

```

@if ($errors->any())
    <div class="alert alert-danger alert-dismissible fade show" role="alert">
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
            <span aria-hidden="true">&times;</span>
        </button>
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif

```

23. Validation:

För CustomerController:

```
$this->validate($request,[
    "person_number" =>
        [
            "required",
            "regex:/^(19|20)?[0-9]{6}[- ]?[0-9]{4}$/i",
            "unique:customers,person_number"
        ],
    "phone_number" => "required|size:10|regex:/^07([0-9][ -]*){7}[0-9]$/i"
]);
```

*(jag fick problem här och sedan har använt array för skriva validation.)

För CarController:

```
$this->validate($request,[
    'registration' =>
        [
            'required',
            'size:6',
            'regex:/^[A-Z a-z]{3}\d{3}$/i',
            "unique:cars,registration"
        ],
    'color' => 'required|exists:colors,id',
    'make' => 'required|exists:makes,id',
    'year' => 'required|numeric|min:1900|max:2020',
    'price' => 'required|integer|min:0'
]);
```

24. Nu behöver jag att ge meddelande till användaren om validationen när han matar in fel:

I resources > lang> validation:

```
'custom' => array(
    'registration' => array(
        'regex' => 'Registration form should be like ABC123'
    ),
```

```

        'person_number' => array(
            'regex' => 'Personnumber form should be like 198808084444 or 8808084444 with
or without dash (-) '
        ),

        'phone_number' => array(
            'regex' => 'The phone number should be like: 0788888888 '
        )
    ),
),

```

25. I view>customer> edit.blade.php (för Car gör jag exakt samma sak): Jag kommer att lägga till *disabled* för att frisera den registration när man ska editera.

```
// $car->registration = $request->registration;
```

```
// $customer->person_number = $request -> person_number;
```

26. Ska man kunna välja mellan färger:

view>car>add.blade.php:

```
<option value="0">--select a color--</option>
```

Men

Man ska i validation kunna välja bara giltiga färger.

I CarController: 'color' => 'required|exists:colors,id'

Och för make också:

view>car>add.blade.php:

```
<option value="0">--select a make--</option>
```

Men

Man ska i validation kunna välja bara giltiga maker.

I CarController: make => 'required|exists:makes,id'

27. När bilen är uthyrd då ska bilens (och kundens) information kunna inte att raderas.
Jag gör så:

```

@if($car->check_out_by == null)
    <a
    href="{{url('car/remove/'. $car->id)}}}"
    class="btn btn-danger btn-sm"
    onclick="return confirm('Are you sure to delet the car?')"
    >Remove</a>
@endif

```

Här knappen kommer att visas när bilen är ledig. Och man blir varnad att ta bort informationen.

Men det är bättre att man fixa det i backend också annars nån kan byta uppgifter. Så:

```
if ($car->check_out_by == null) {  
  
    $histories = History::where('car_id', $id)->get();  
    foreach($histories as $history)  
        $history->delete();  
  
    $car->delete();  
}  
  
return redirect('cars')  
    ->with('message', 'The information of the car deleted successfully.');
```

Nu det finns mer säkerhet och det kommer att ta bort informationen från History också. Dessutom användaren får meddelande.

För Customer gör jag samma sak.

28. Att fixa tiden med Stockholm tid som står på history, man fixar det i Config:

```
Config> App 'timezone' => 'Europe/Stockholm'
```