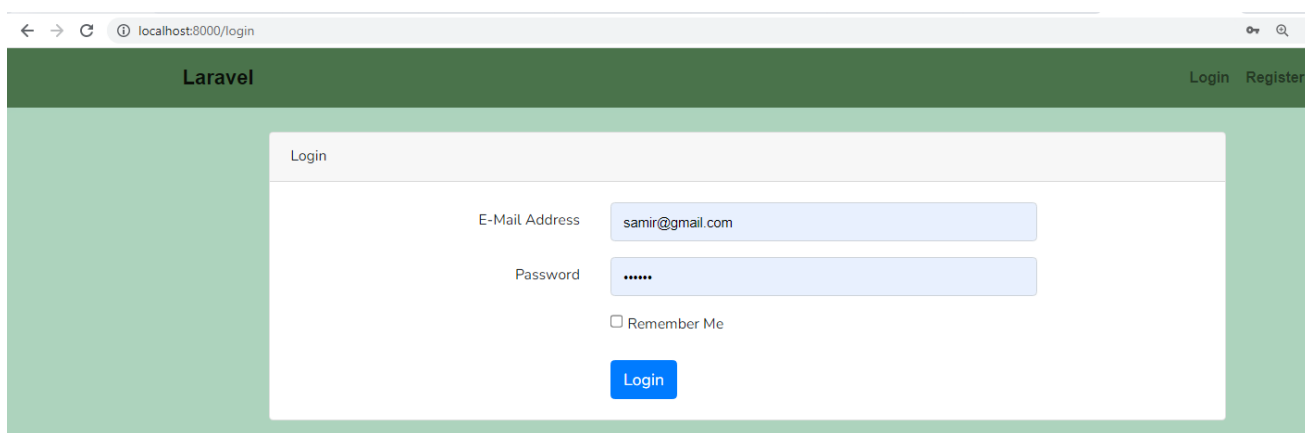


Authentification et inscription personnalisées dans Laravel:

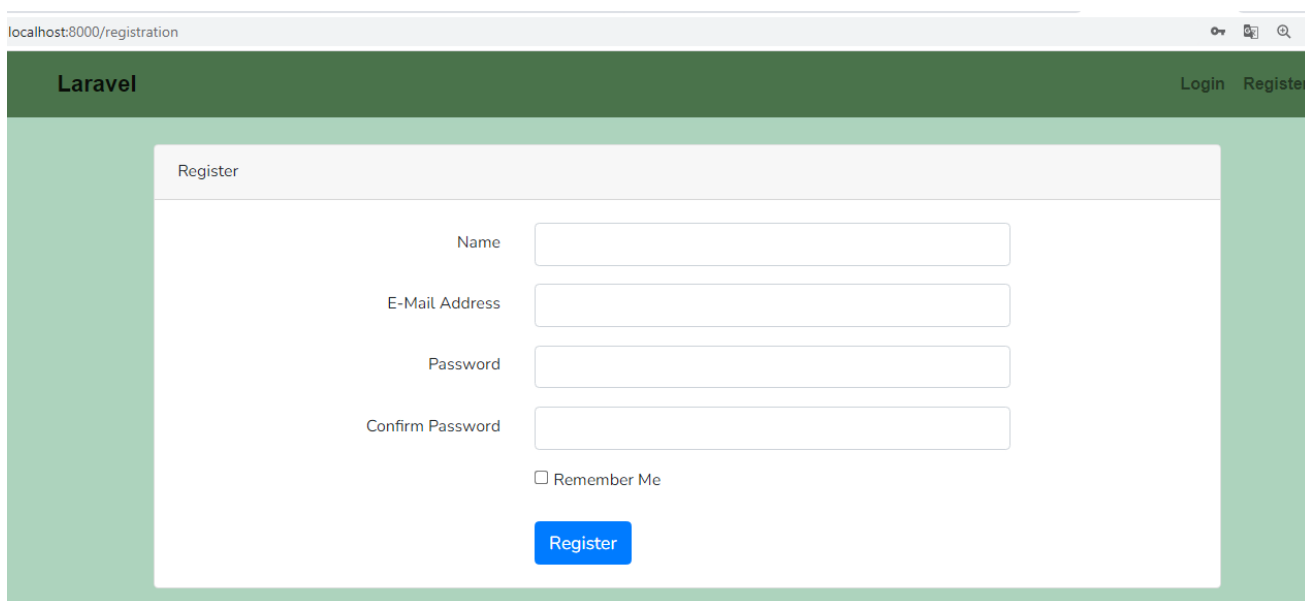
Laravel fournit l'authentification en utilisant **jetstream** et le package **UI**. Mais parfois nous avons besoin de créer notre propre **login**, enregistrement, tableau de bord, et **logout** alors on va créer étape par étape un login personnalisé et une page d'enregistrement dans l'application Laravel.

Login Page:



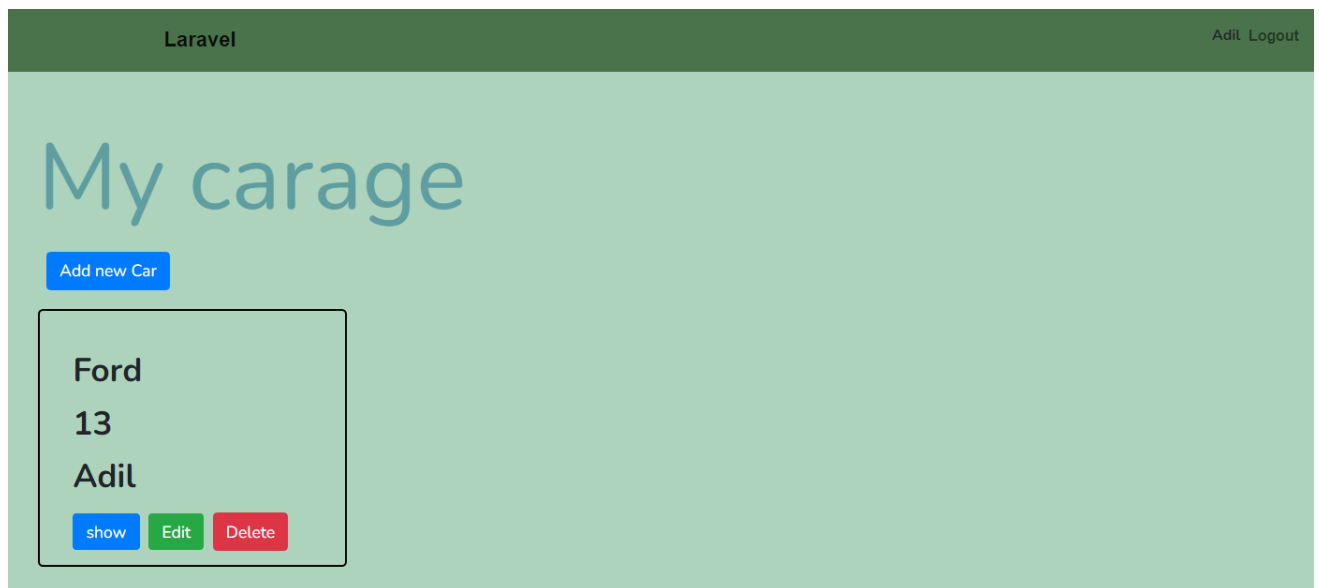
The screenshot shows a web browser at the URL `localhost:8000/login`. The page has a dark green header with the text "Laravel" on the left and "Login Register" on the right. The main content area has a light green background. In the center, there is a white box titled "Login". Inside this box, there are two input fields: "E-Mail Address" with the value "samir@gmail.com" and "Password" with masked characters "*****". Below these fields is a checkbox labeled "Remember Me" which is unchecked. At the bottom of the box is a blue button labeled "Login".

Register Page:



The screenshot shows a web browser at the URL `localhost:8000/registration`. The page has a dark green header with the text "Laravel" on the left and "Login Register" on the right. The main content area has a light green background. In the center, there is a white box titled "Register". Inside this box, there are four input fields: "Name", "E-Mail Address", "Password", and "Confirm Password". Below these fields is a checkbox labeled "Remember Me" which is unchecked. At the bottom of the box is a blue button labeled "Register".

Index Page:



Etape 1: Configuration de la base de données

Tout d'abord, nous devons obtenir une nouvelle version de l'application Laravel en utilisant la commande ci-dessous. Ouvrez votre terminal ou votre invite de commande et exécutez la commande ci-dessous :

```
composer create-project laravel/laravel example-app
```

Ensuite, nous devons ajouter la configuration de la base de données dans le fichier `.env` . Ajoutons donc les détails suivants et exécutons ensuite la commande de migration :

.env

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=myDB
DB_USERNAME=root
```

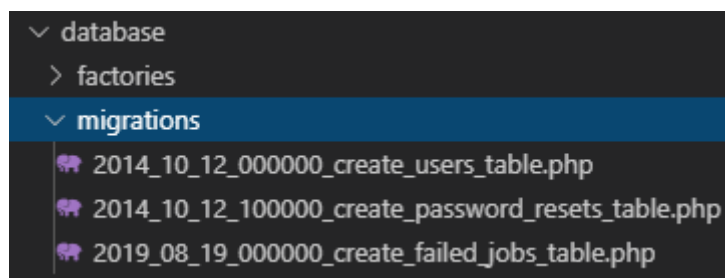
```
DB_PASSWORD=
```

Ensuite, exécutez la commande de migration pour créer la table `users` .

```
php artisan migrate
```

Par défaut la partie persistance de l'authentification (c'est à dire la manière de retrouver les renseignements des utilisateurs) avec Laravel se fait en base de données avec Eloquent et part du principe qu'il y a un modèle **App\Models\User** (dans le dossier app).

Lors de l'installation on a vu dans les chapitres précédents qu'il existe déjà des migrations :



Repartez d'une installation vierge et faites la migration avec Artisan :

```
λ php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (375.60ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (347.34ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (359.37ms)
```

Vous devriez normalement obtenir ces tables :

- table **users** : par défaut Laravel considère que cette table existe et il s'en sert comme référence pour l'authentification.
- table **password_resets** : cette table va nous servir pour la réinitialisation des mots de passe.

Etape 2: Configuration du modèle User et Création du modèle Car

//App\Models\Car.php

```
class Car extends Model
{
    use HasFactory;
    public function user(){

        return $this->belongsTo(User::class);

    }
    protected $fillable = [
        'brand',
        'price',
        'picture',
        'origin'
    ];
}
```

//App\Models\User.php

```
<?php

namespace App\Models;

use App\Models\Car;
use Illuminate\Notifications\Notifiable;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable implements MustVerifyEmail
{
    use HasFactory, Notifiable;
    public function cars(){
        return $this->hasMany(Car::class);
    }
    public function userverify(){
```

```

        return $this->hasOne(UserVerify::class);
    }

    protected $fillable = [
        'name',
        'email',
        'password',
    ];

    protected $hidden = [
        'password',
        'remember_token',
    ];

    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}

```

database/migrations/xxxx_create_cars_table.php

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('cars', function (Blueprint $table) {
            $table->id();
            $table->string('brand');
            $table->integer('price');
            $table->string('origin');
        });
    }
}

```

```

        $table->string('picture');
        $table->unsignedBigInteger('user_id');
        $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('cars');
}
};

```

Etape 3: Créer le contrôleur d'Authentification

Dans cette étape, nous devons créer **AuthController** et ajouter le code suivant dans ce fichier :

app/Http/Controllers/Auth/AuthController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Illuminate\Auth\Events\Registered;
use Illuminate\Support\Facades\Session;
use Illuminate\Support\Facades\Validator;

class AuthController extends Controller
{
    public function index()
    {
        return view('auth.login');
    }
}

```

```

public function registration()
{
    return view('auth.registration');
}

public function postLogin(Request $request)
{
    $rules = [
        'email' => ['required', 'email'],
        'password' => ['required', 'min:6'],
    ];
    $messages=[
        'email.email'=>'email not valid'
    ];

    $validateForm=Validator::make($request->all(),$rules,$messages);
    if($validateForm->fails())
    return redirect()->back()->withErrors($validateForm)->onlyInput('email');

    if (Auth::attempt(array('email'=>$request->email,'password'=>$request->password),$request->has('remember')) {
        $request->session()->regenerate();

        return redirect()->intended('cars');
    }

    return redirect("login")->withSuccess('Oppes! You have entered invalid credentials')->onlyInput('email');
}

public function postRegistration(Request $request)
{
    $rules=[
        'name' => 'required',
        'email' => 'required|email|unique:users',
        'password' => 'required|min:6',
        'confirmPassword' => 'required|same:password',
    ];

    $validateData=Validator::make($request->all(),$rules);
    if($validateData->fails())
    return redirect()->back()->withErrors($validateData)->withInput();
    $data=[
        "name"=>$request->name,
        "password"=>$request->password,
        "email"=>$request->email,
    ];
}

```

```

        $user=$this->create($data);
        Auth::attempt(array('email'=>$data['email'],'password'=>$data['password']));
        return redirect("/cars")->withSuccess('Great! You have Successfully
loggedin');
    }

    public function create(array $data)
    {

        return User::create([
            'name' => $data['name'],
            'email' => $data['email'],
            'password' => Hash::make($data['password'])
        ]);

    }

    public function logout() {
        Session::flush();
        Auth::logout();

        return Redirect('login');
    }
}

```

La méthode **attempt** renvoie **true** si l'authentification a réussi. Dans le cas contraire, false sera retourné.

La méthode **intended** fournie par le redirecteur de Laravel redirigera l'utilisateur vers l'URL à laquelle il tentait d'accéder avant d'être intercepté par le middleware d'authentification. Une URI de repli peut être donnée à cette méthode au cas où la destination prévue ne serait pas disponible.

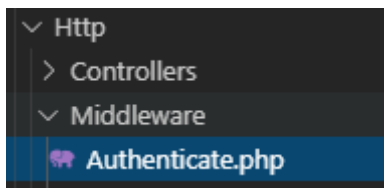
Etape 4 : Les middlewares

Je vous ai déjà parlé des middlewares qui servent à effectuer un traitement à l'arrivée (ou au départ) des requêtes HTTP. On a vu le middleware de groupe **web** qui intervient pour toutes les requêtes qui arrivent. Dans ce TP on va voir deux autres middlewares qui vont nous permettre de savoir si un utilisateur est authentifié ou pas pour agir en conséquence.

Middleware auth

Le middleware **auth** permet de n'autoriser l'accès qu'aux utilisateurs authentifiés. Ce middleware est déjà présent et déclaré dans **app\Http\Kernel.php** :

```
protected $routeMiddleware = [  
'auth' => \App\Http\Middleware\Authenticate::class,  
...  
];
```



On peut utiliser ce middleware directement sur une route :

```
Route::get('comptes', function() {  
    // Réserve aux utilisateurs authentifiés  
})->middleware('auth');
```

Ou un groupe de routes :

```
Route::middleware('auth')->group(function () {  
    Route::get('/', function () {  
        // Réserve aux utilisateurs authentifiés  
    });  
    Route::get('comptes', function () {  
        // Réserve aux utilisateurs authentifiés  
    });  
});
```

Middleware guest

Ce middleware est exactement l'inverse du précédent : il permet de n'autoriser l'accès qu'aux utilisateurs non authentifiés. Ce middleware est aussi déjà présent et déclaré dans `app\Http\Kernel.php` :

```
protected $routeMiddleware = [  
  
...  
  
'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,  
  
...  
  
];
```

De la même manière que **auth**, le middleware **guest**, comme d'ailleurs tous les middlewares, peut s'utiliser sur une route, un groupe de routes ou dans le constructeur d'un contrôleur, avec la même syntaxe.

Etape 5: la protection des routes

Dans cette étape, nous devons créer une route personnalisée pour la connexion, l'enregistrement, l'accueil et la déconnexion. Ouvrez donc votre fichier `routes/web.php` et ajoutez les routes suivantes.

routes/web.php

```
//Car Route  
Route::middleware(['auth'])->group(function () {  
    Route::resource('cars', CarController::class);  
    Route::get('logout', [AuthController::class, 'logout'])->name('logout');  
});  
  
//Auth Routes  
  
Route::group(['middleware'=>'guest'],function(){  
Route::get('/',function(){  
    return redirect()->route('login');  
});  
Route::get('login', [AuthController::class, 'index'])->name('login');  
Route::post('post-login', [AuthController::class, 'postLogin'])->name('login.post');  
Route::get('registration', [AuthController::class, 'registration'])->  
>name('register');
```

```
Route::post('post-registration', [AuthController::class, 'postRegistration'])-
>name('register.post');

});
```

Comme vous pouvez le voir ci-dessus dans cette ligne "**Route::group(['middleware' => ['auth']], function() {"** nous avons protégé la route de déconnexion qui ne peut être accessible que si l'utilisateur est authentifié. Si vous avez d'autres routes à protéger, ajoutez simplement ce groupe de routes.

Etape 6: Créer les vues en Blade

Ici, nous devons créer des fichiers de blade pour `layout`, `login`, `register` et `index` page. Donc créons un par un les fichiers :

resources/views/auth/login.blade.php

```
@extends('layouts.layout')

@section('content')
<main class="login-form">
  <div class="cotainer">
    <div class="row justify-content-center">
      <div class="col-md-8">
        <div class="card">
          <div class="card-header">Login</div>
          <div class="card-body">

            <form action="{{ route('login.post') }}" method="POST">
              @csrf
              <div class="form-group row">
                <label for="email_address" class="col-md-4 col-form-label text-md-right">E-Mail Address</label>
                <div class="col-md-6">
                  <input type="text" id="email_address" class="form-control @error('email') border border-danger @enderror" name="email"
value={{old('email')}} >
                  @error('email')
                    <span class="text-danger">{{ $message}}</span>
                  @enderror
                </div>
              </div>
            </div>

            <div class="form-group row">
```

```

        <label for="password" class="col-md-4 col-form-label
text-md-right">Password</label>
        <div class="col-md-6">
            <input type="password" id="password" class="form-
control @error('password') border border-danger @enderror" name="password" >
                @error('password')
                <span class="text-danger">{{ $message}}</span>
            @enderror
        </div>
    </div>

    <div class="form-group row">
        <div class="col-md-6 offset-md-4">
            <div class="checkbox">
                <label>
                    <input type="checkbox" name="remember">
Remember Me
                </label>
            </div>
        </div>
    </div>

    <div class="col-md-6 offset-md-4">
        <button type="submit" class="btn btn-primary">
Login
        </button>
    </div>
    @if(Session::has('success'))
    <div class="alert alert-danger">
        {{Session::get('success')}}
    </div>
    @endif

    </form>

    </div>
</div>
</div>
</div>
</div>
</main>
@endsection

```

resources/views/auth/registration.blade.php

```
@extends('layouts.layout')
```

```
@section('content')
```

```

<main class="login-form">
  <div class="container">
    <div class="row justify-content-center">
      <div class="col-md-8">
        <div class="card">
          <div class="card-header">Register</div>
          <div class="card-body">

            <form action="{{ route('register.post') }}" method="POST">
              @csrf
              <div class="form-group row">
                <label for="name" class="col-md-4 col-form-label text-
md-right">Name</label>

                <div class="col-md-6">
                  <input type="text" id="name" class="form-control
@error('name') border border-danger @enderror" name="name" value="{{old('name')}}" >
                  @error('name')
                    <span class="text-danger">{{ $message}}</span>
                  @enderror
                </div>
              </div>

              <div class="form-group row">
                <label for="email_address" class="col-md-4 col-form-
label text-md-right">E-Mail Address</label>
                <div class="col-md-6">
                  <input type="text" id="email_address" class="form-
control @error('email') border border-danger @enderror" name="email"
value="{{old('email')}}" >
                  @error('email')
                    <span class="text-danger">{{ $message}}</span>
                  @enderror
                </div>
              </div>

              <div class="form-group row">
                <label for="password" class="col-md-4 col-form-label
text-md-right">Password</label>
                <div class="col-md-6">
                  <input type="password" id="password" class="form-
control @error('password') border border-danger @enderror" name="password"
value="{{old('password')}}" >
                  @error('password')
                    <span class="text-danger">{{ $message}}</span>
                  @enderror
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <label for="confirmPassword" class="col-md-4 col-form-label text-md-right">Confirm Password</label>
        <div class="col-md-6">
            <input type="password" id="confirmPassword"
class="form-control @error('confirmPassword') border border-danger @enderror"
name="confirmPassword">
            @error('confirmPassword')
            <span class="text-danger">{{ $message}}</span>
            @enderror
        </div>
    </div>

    <div class="form-group row">
        <div class="col-md-6 offset-md-4">
            <div class="checkbox">
                <label>
                    <input type="checkbox" name="remember">
Remember Me
                </label>
            </div>
        </div>
    </div>

    <div class="col-md-6 offset-md-4">
        <button type="submit" class="btn btn-primary">
            Register
        </button>
    </div>
</form>

</div>
</div>
</div>
</div>
</main>
@endsection

```

resources/views/layouts/layout.blade.php

```

<!DOCTYPE html>
<html>
<head>
    <title>Laravel</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
    <style type="text/css">
        @import url(https://fonts.googleapis.com/css?family=Raleway:300,400,600);
    </style>
</head>

```

```

body{
    margin: 0;
    font-size: .9rem;
    font-weight: 400;
    line-height: 1.6;
    color: #212529;
    text-align: left;
    background-color: #f5f8fa;
}
.navbar-laravel
{
    box-shadow: 0 2px 4px rgba(0,0,0,.04);
}
.navbar-brand , .nav-link, .my-form, .login-form
{
    font-family: Raleway, sans-serif;
}
.my-form
{
    padding-top: 1.5rem;
    padding-bottom: 1.5rem;
}
.my-form .row
{
    margin-left: 0;
    margin-right: 0;
}
.login-form
{
    padding-top: 1.5rem;
    padding-bottom: 1.5rem;
}
.login-form .row
{
    margin-left: 0;
    margin-right: 0;
}
}
</style>
</head>
<body>

<nav class="navbar navbar-expand-lg navbar-light navbar-laravel">
    <div class="container">
        <a class="navbar-brand" href="{{route('students.index')}}">Laravel</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>

```

```

        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav ml-auto ">
                @guest
                    <li class="nav-item ">
                        <a class="nav-link" href="{{ route('login') }}">Login</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ route('register')
}}">Register</a>
                    </li>
                @else
                    <li class="nav-item mt-2">
                        <p>{{Auth::user()->name}}</p>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ route('logout') }}">Logout</a>
                    </li>
                @endguest
            </ul>

        </div>
    </div>
</nav>

@yield('content')

</body>
</html>

```

resources/views/cars/index.blade.php

```

@extends('layouts.layout')
@section('title',"cars")
@section('content')

    <div class="carscontainer">
        @if (session('success'))
            <div class="alert alert-success">{{Session::get('success')}}</div>

        @endif
        <h1>My carage</h1>

        <a href="{{route('cars.create')}}" class="btn btn-primary m-2">Add new Car</a>
        <div class="cars">
            @forelse ($cars as $car)

```



```

<div class="car">
  <span> {{ $car['brand'] }}</span>
  <span>{{ $car['price'] }}</span>
  {{-- <span>{{ $item['origin'] }}</span> --}}
  <span>{{ $car['user']->name }}</span>
  <div class='groupbtn'>

<form action="{{ route('cars.destroy', $car) }}" method="POST">
  @method('DELETE')
  @csrf
  <a href="{{ route('cars.show', $car) }}" class="btn btn-primary"> show</a>
  <a href="{{ route('cars.edit', $car) }}" class="btn btn-success">Edit</a>
  @can('delete', $car)
    <button class="btn btn-danger">Delete</button>
  @endcan

</form>
  </div>

</div>

@empty
  <h3>there is no cars</h3>

@endforelse

</div></div>
@endsection

```

Laravel

My carage

Add new Car

Renault

23236

Hiba

show

Edit

Delete

Mercedes

12

Hiba

show

Edit

Delete