

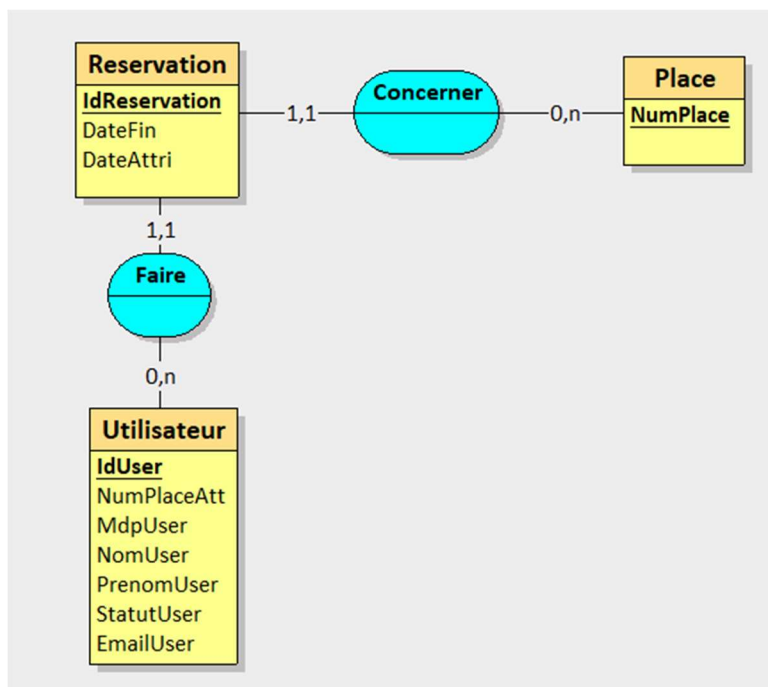
# Documentation du projet Parking

## Sommaire :

- **Base de données**
  - MCD (langage SQL, et SGBD MySQL)
  - Script
- **Code Laravel**

## Base de données

### MCD



Un utilisateur peut réserver des places (une à la fois), et une réservation concerne un utilisateur et une seule place.

## Script :

```
CREATE TABLE Utilisateur(  
    IdUser INT,  
    NumPlaceAtt INT,  
    MdpUser VARCHAR(50),  
    NomUser VARCHAR(50),  
    PrenomUser VARCHAR(50),  
    StatutUser VARCHAR(20),  
    EmailUser VARCHAR(25),  
    PRIMARY KEY (IdUser)  
)ENGINE =INNODB;  
  
CREATE TABLE Reservation(  
    IdReservation INT,  
    IdUser INT,  
    NumPlace INT,  
    DateFin DATE,  
    DateAttribution DATE,  
    PRIMARY KEY (IdReservation)  
)ENGINE = INNODB;  
  
CREATE TABLE Place(  
    NumPlace INT,  
    StatutPlace VARCHAR(20),  
    PRIMARY KEY (NumPlace)  
)ENGINE = INNODB;
```

```
ALTER TABLE Reservation  
DROP CONSTRAINT FK_RES_PLA;  
ALTER TABLE Reservation  
DROP CONSTRAINT FK_RES_UTI;  
  
ALTER TABLE Reservation  
ADD CONSTRAINT FK_RES_UTI FOREIGN KEY (IdUser) REFERENCES Utilisateur (IdUser);  
  
ALTER TABLE Reservation  
ADD CONSTRAINT FK_RES_PLA FOREIGN KEY (NumPlace) REFERENCES Place (NumPlace);
```

Contrainte assurant les liens entre réservation et utilisateur/place.

# Code Laravel

```
<?php

use Illuminate\Support\Facades\Route;
require __DIR__.'/auth.php';

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

/*Route::get('/', function () {
    return view('welcome');
});*/

Route::get(uri: '/', function () { // renvoie la vue apres l'authentification du middleware
    return view(view: 'parking.accueil');
})->middleware(['auth'])->name(name: 'dashboard'); //Authentification dans la vue dashboard

Route::get(uri: '/accueil', function () {
    return view(view: 'parking.accueil');
});

Route::get(uri: '/etatreservation', function () {
    return view(view: 'parking.etatreservation');
});

Route::get(uri: '/anciennereservation', function () {
    return view(view: 'parking.anciennereservation');
});

Route::get(uri: '/reserver', action: 'ReservationController@store')->name(name: 'reservations.store');
```

Les routes de notre application

```

index.blade.php x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Liste des réservations</title>
5 </head>
6 <body>
7 <h1>Liste des réservations</h1>
8 <table>
9 <thead>
10 <tr>
11 <th>Id</th>
12 <th>Statut</th>
13 </tr>
14 </thead>
15 <tbody>
16 @foreach ($places as $places)
17 <tr>
18 <td>{{ $places->Id }}</td>
19 <td>{{ $places->StatutPlace }}</td>
20 </tr>
21 @endforeach
22 </tbody>
23 </table>
24 </body>
25 </html>

```

Ce fichier affiche  
affiche l'id de la place, ainsi que son statut.

```

reserver.blade.php x
35 background-color: rgb(123, 123, 196);
36 width: 200px;
37 height: 50px;
38 }
39 button:hover{
40 cursor:pointer;
41 }
42 </style>
43 </head>
44 <body>
45
46 <div class = "Container FadeInEffect">
47
48 <div class = "NavigationBarWrapper">
49
50 <nav class = "FullNavigationBar">
51
52 <a href="ParkingAccueil" id = "accueil">Accueil</a>
53 <a href="reserver" id = "reserver">Reserver</a>
54 <a href="etatreservation" id = "etatreservation">Etat Reservation</a>
55 <a href="anciennereservation" id = "anciennereservation">Ancienne Reservation</a>
56 </nav>
57
58 </div>
59
60 <!-- <a href="{{ route('dashboard') }}" id = "accueil">Accueil</a>
61 <a href="{{ route('reserver') }}" id = "reserver">Reserver</a>
62 <a href="{{ route('etatreservation') }}" id = "etatreservation">Etat Reservation</a>
63 <a href="{{ route('anciennereservation') }}" id = "anciennereservation">Ancienne Reservation</a-->
64
65 <div class = "buttonposition">
66 <form action = "/reservations" method = "GET">
67 <button type="submit">Réservé</button>
68 </form>
69 </div>
70
71 </div>
72
73 </body>
74 </html>
75
76

```

De même pour réserver,

```
welcome.blade.php
1 |<!DOCTYPE html>
2 |<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3 |<head>
4 |   <meta charset="utf-8">
5 |   <meta name="viewport" content="width=device-width, initial-scale=1">
6 |
7 |   <title>Laravel</title>
8 |
9 |   <!-- Fonts -->
10 |  <link href="https://fonts.googleapis.com/css2?family=Nunito:wght@400;600;700&display=swap" rel="stylesheet">
11 |
12 |  <!-- Styles -->
13 |  <style>
14 |    /*! normalize.css v8.0.1 | MIT License | github.com/necolas/normalize.css */<html{line-height:1.15;-webkit-text-size-adjust:100%;}bo
15 |  </style>
16 |
17 |  <style>
18 |    body {
19 |      font-family: 'Nunito', sans-serif;
20 |    }
21 |  </style>
22 |</head>
23 |<body class="antialiased">
24 |  <div class="relative flex items-top justify-center min-h-screen bg-gray-100 dark:bg-gray-900 sm:items-center py-4 sm:pt-0">
25 |    @if (Route::has('login'))
26 |      <div class="hidden fixed top-0 right-0 px-6 py-4 sm:block">
27 |        @auth
28 |          <a href="{{ url('/dashboard') }}" class="text-sm text-gray-700 dark:text-gray-500 underline">Dashboard</a>
29 |        @else
30 |          <a href="{{ route('login') }}" class="text-sm text-gray-700 dark:text-gray-500 underline">Log in</a>
31 |
32 |          @if (Route::has('register'))
33 |            <a href="{{ route('register') }}" class="ml-4 text-sm text-gray-700 dark:text-gray-500 underline">Register</a>
34 |          @endif
35 |        @endauth
36 |      </div>
37 |    @endif
38 |
39 |    <div class="max-w-6xl mx-auto sm:px-6 lg:px-8">
40 |      <div class="flex justify-center pt-8 sm:justify-start sm:pt-0">
41 |        <svg viewBox="0 0 651 192" fill="none" xmlns="http://www.w3.org/2000/svg" class="h-16 w-auto text-gray-700 sm:h-20">
42 |          <g clip-path="url(#clip0)" fill="#EF3820">
43 |            <path d="M248.032 44.676h-16.466v100.23h47.394v-14.748h-30.928V44.676zM337.091 87.202c-2.101-3.341-5.083-5.965-8.94
44 |          </g>
45 |        </svg>
46 |      </div>
47 |    </div>
48 |  </div>
```

Ici, on représente la vue principale de l'application

Ensuite, nous avons les migrations de notre application, avec :

```
php 2014_10_12_000000_create_users_table.php x
1  <?php
2
3  > use ...
4
5  namespace
6
7  class CreateUsersTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14       public function up()
15       {
16           Schema::create( 'users', function (Blueprint $table) {
17               $table->id();
18               $table->string( 'column: 'name');
19               $table->string( 'column: 'email')->unique();
20               $table->timestamp( 'column: 'email_verified_at')->nullable();
21               $table->string( 'column: 'password');
22               $table->integer( 'column: 'NumPlaceAtt')->nullable();
23               $table->string( 'column: 'PrenomUser', length: 50);
24               $table->string( 'column: 'StatutUser', length: 20)->default( 'value: 0');
25               $table->rememberToken();
26               $table->timestamps();
27           });
28       }
29
30       /**
31       * Reverse the migrations.
32       *
33       * @return void
34       */
35       public function down()
36       {
37           Schema::dropIfExists( 'users');
38       }
39   }
40
```

La migration créant la table utilisateurs ( users),

```
© 2024_03_04_164035_reservation.php x
1  <?php
2
3  > use ...
6
   migrations
7  class Reservation extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create( table: 'reservations', function(Blueprint $table) {
17             $table->id();
18             $table->foreign( columns: 'idUser')->references( columns: 'id')->on( table: 'users');
19             $table->integer( column: 'NumPlace');
20             $table->foreign( columns: 'NumPlace')->references( columns: 'id')->on( table: 'Places');
21             $table->date( column: 'DateFin');
22             $table->date( column: 'DateAttribution');
23         });
24     }
25     /**
26      * Reverse the migrations.
27      *
28      * @return void
29      */
30     public function down()
31     {
32         Schema::drop( table: 'Reservations');
33     }
34 }
35
```

La migration créant la table réservation,

Et enfin la migration créant la table Place :

```
© 2024_03_04_164041_place.php ×
1  <?php
2
3  > use ...
6
   no usages
7  class Place extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create( table: 'Places', function(Blueprint $table) {
17             $table->integer( column: 'Id');
18             $table->string( column: 'StatutPlace', length: 20);
19         });
20     }
21
22     /**
23      * Reverse the migrations.
24      *
25      * @return void
26      */
27     public function down()
28     {
29         Schema::drop( table: 'Places');
30     }
31 }
32
```



