

**Laravel**

# Agenda

- Introduction
- Laravel features
- Exercises
- Questionnaire
- Wrap-up



# What is Laravel?

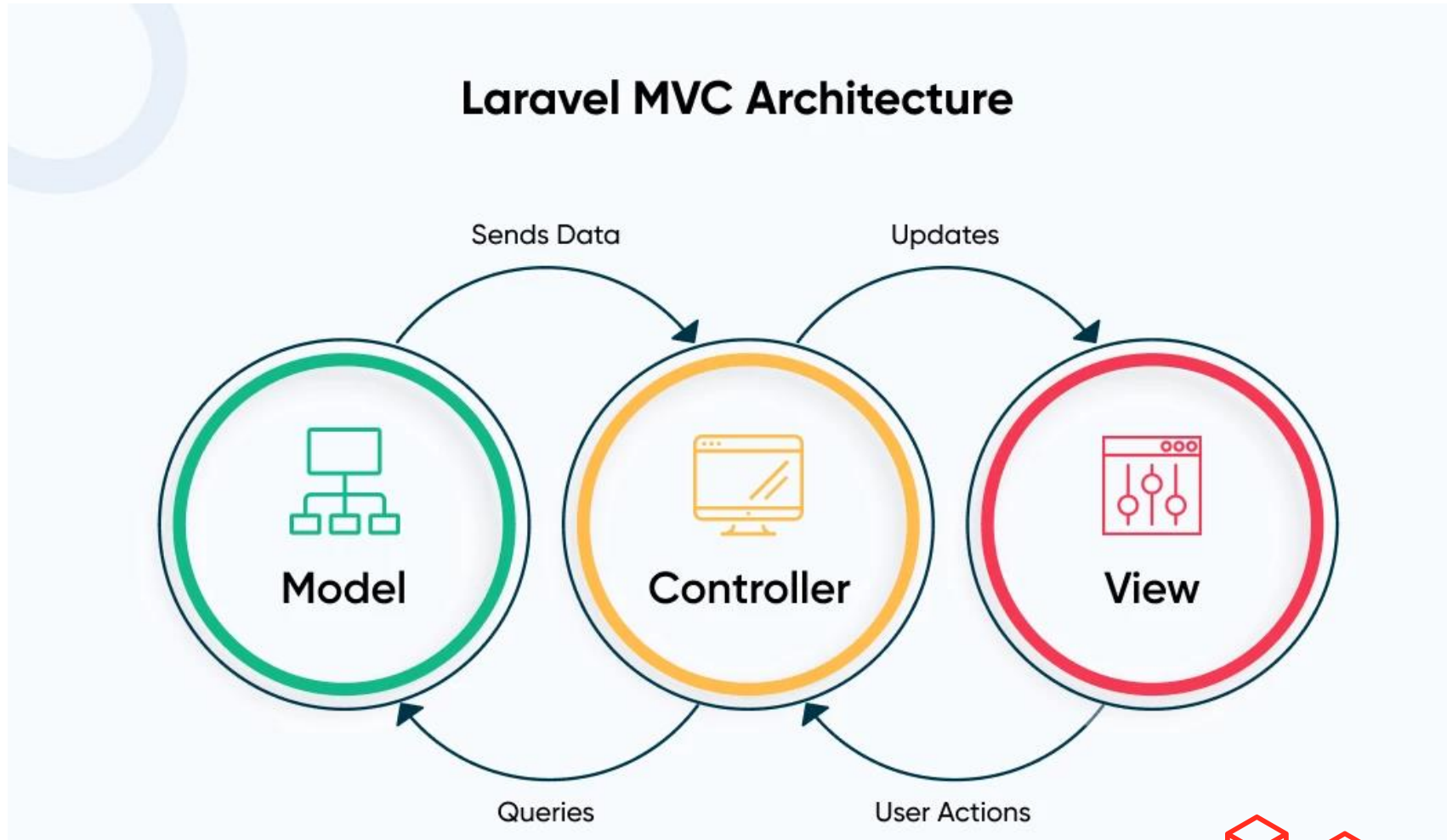
- Web-Framework
- PHP-Based
- Artisan

Basics:

- Authentication
- Database
- Queues



# Laravel architecture



# Migrations

- Version control
- Extend
- Modify
- Share

```
php artisan make:migration create_cars_table
```

>

```
php artisan migrate
```

```
INFO Running migrations.
```

```
2022_11_16_083702_create_cars_table .....  
..... 317ms DONE
```

```
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->string('type');  
            $table->string('mileage')->nullable();  
            $table->string('color');  
            $table->float('price');  
            $table->boolean('sold');  
            $table->timestamps();  
        });  
    }  
  
    /**  
     * Reverse the migrations.  
     *  
     * @return void  
     */  
    public function down()  
    {  
        Schema::dropIfExists('cars');  
    }  
};
```



# Models

- MVC
- Template
- Questioning the database
- Eloquent

```
php artisan make:model Car
```

>

```
INFO Model [app/Models/Car.php] created successfully.
```

```
class Car extends Model
{
    use HasFactory;

    //Appends getCarNameAttribute to the model
    protected $appends = ['car_name'];

    //Casting sold tiny-int to boolean
    protected $casts = [
        'sold' => 'boolean',
    ];

    //Creates a new Attribute combining brand and type
    public function getCarNameAttribute(){
        return $this->brand . ' ' . $this->type;
    }

    //Formatting a date to proper europe time format
    public function getCreatedAtAttribute($date) {
        return Carbon::createFromDate($date)->format('d-m-Y H:i');
    }
}
```

# Routes

- Links
- Demands
- Controllers

```
Route::middleware([
    'auth:sanctum',
    config('jetstream.auth_session'),
    'verified',
])->group(function () {
    Route::get('/dashboard', function () {
        return Inertia::render('Dashboard');
    })->name('dashboard');

    Route::get('/cars', [Carcontroller::class, 'display'])
        ->name('cars.display');
});
```

# Controllers

- Handle Requests
- Handle Logic

```
php artisan make:controller CarController >
```

```
INFO Controller [app/Http/Controllers/CarController.php] created successfully.
```

```
class CarController extends Controller
{
    //Retrieves all cars in database and renders the Dashboard with cars as props
    public function display(){
        $cars = Car::all();
        return Inertia::render('Car/Dashboard',[
            'cars' => $cars
        ]);
    }
}
```



# Eloquent

- Querying
- Modifying
- SQL databases
- Transactions

```
$cars = Car::all();  
$carsSearched = Car::find(1);  
$carsWhere = Car::where('brand', '=', 'Volkswagen')->get();  
$carsWithOwner = Car::all()->with('owner');  
$carsOrderedByDesc = Car::all()->orderByDesc('id');
```

```
DB::transaction(function() use ($request){  
    $car = new Car(); //Creates a new model  
    $car->brand = $request->get('brand'); //Set value to model  
    $car->type = $request->get('type');  
    $car->color = $request->get('color');  
    $car->save(); //Saves the model to data  
});
```

Documentation <https://laravel.com/docs/9.x/eloquent>



# Validation

- Input validation
- Error Response

```
php artisan make:request StoreCarRequestValidation >
```

```
INFO Request [app/Http/Requests/StoreCarRequestValidation.php] created successfully.
```

```
class StoreCarRequestValidation extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array<string, mixed>
     */
    public function rules()
    {
        return [
            'brand' => [
                'required',
                'string',
            ],
            'type' => [
                'required',
                'string',
            ],
        ];
    }
}
```



# Inertia

- Rendering
- Transmitting data

```
$cars = Car::all();  
return Inertia::render('Car/Dashboard', [  
    'cars' => $cars  
]);
```



# Scheduler and Commands

- Schedule Kernel.php

```
protected function schedule(Schedule $schedule)
{
    $schedule->command('inspire')->hourly();
}
```

- Triggers > Console.php

```
Artisan::command('inspire', function () {
    $this->comment(Inspiring::quote());
})->purpose('Display an inspiring quote');
```

A scene from the Pixar movie 'Cars' showing Lightning McQueen (a red sports car) and Doc Hudson (a blue classic car) driving on a dirt road through a desert landscape with red rock formations in the background. The word 'Exercise' is overlaid in white text.

# Exercise

In the supplied project you will find an application about managing cars together with a list of exercises

# Wrap-up

- Small questionnaire on today's topic
- <https://kahoot.it/>