



Date

18th May 2024 – 19th May 2024

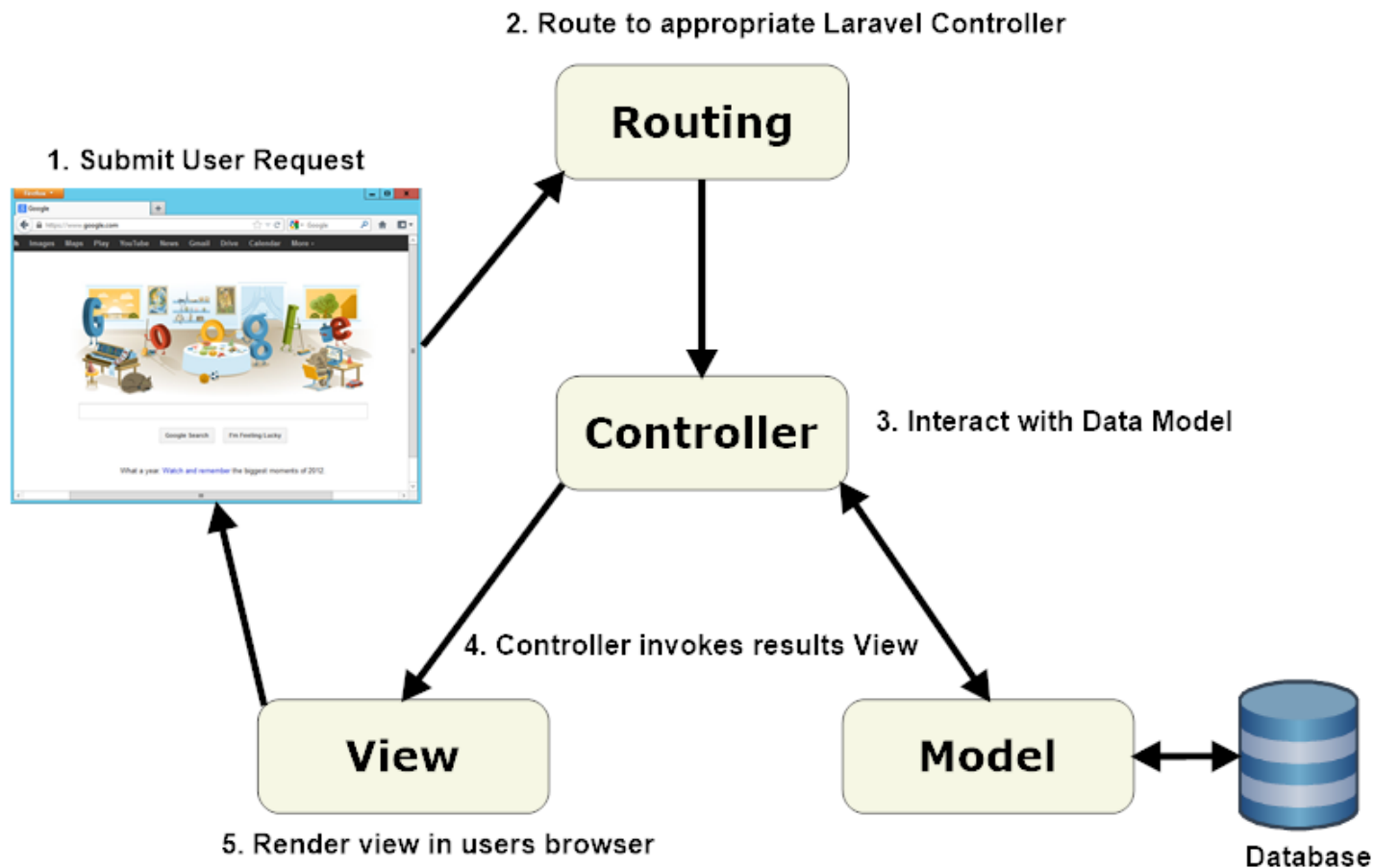
Trainer

AHMAD SAIFUL BAHRI BIN MAT AMIN

LARAVEL

- **Laravel is a free, open-source PHP web framework**
- **Created by Taylor Otwell**
- **Intended for the development of web applications**
- **Following the model–view–controller (MVC) architectural**
- **Current version is Laravel 10.X**

Model – View – Controller (MVC)





02

SETUP & INSTALLATION

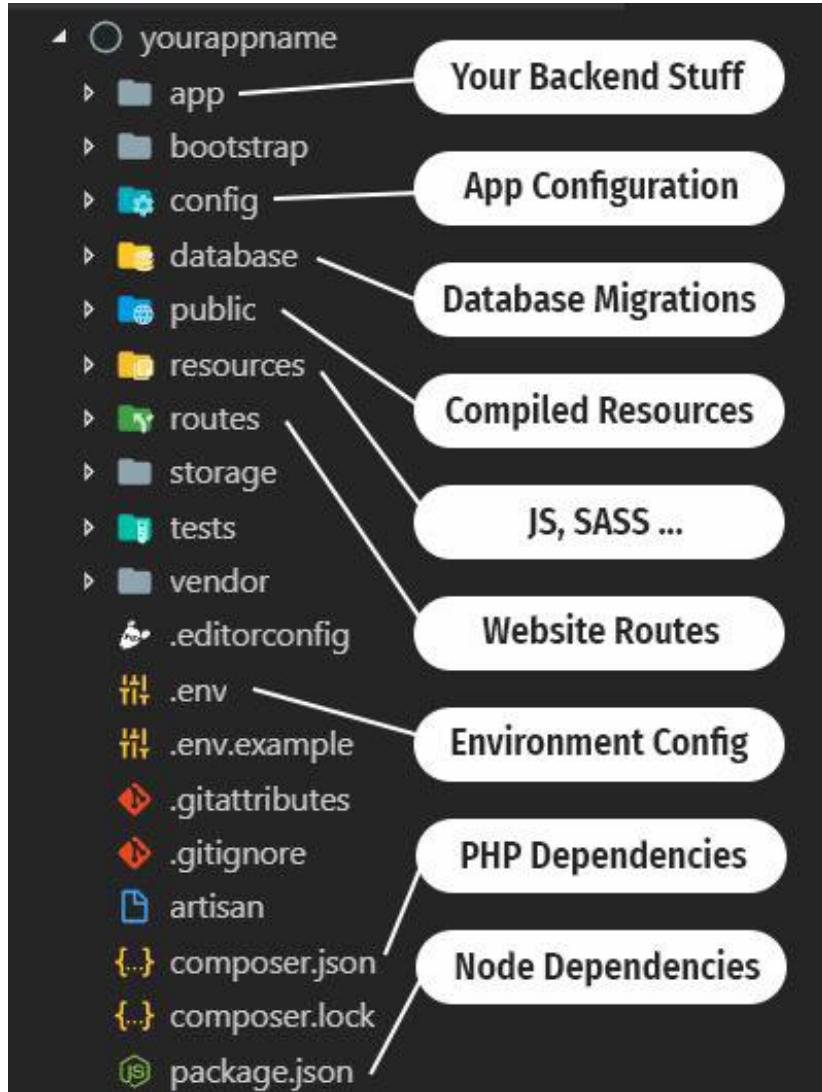
Download & Install Laragon (Windows) – <https://laragon.org>

Using Composer Create Project

```
$ composer create-project --prefer-dist laravel/laravel:^9.x <project_name>
```

Or Using Laravel Installer

```
$ composer global require laravel/installer  
$ laravel new <project_name>
```



Laravel project folder structure

APP_NAME=<project_name>

APP_ENV=<environment: local/production>

APP_KEY=base64:0iRg2ydvJDhJ9uAu2xld3hCTJ/XWAHSU/DTQrsVyteo=

APP_DEBUG=<debug: true/false>

APP_URL=https://<project_url>

DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=<database_name>

DB_USERNAME=<database_username>

DB_PASSWORD=<database_password>

Route Location: **routes/web.php**

return content directly

```
Route::get('/', function() {  
    return "hello world!";  
});
```

return view

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

return controller

```
Route::get('/profile', [  
    App\Http\Controllers\ProfileController::class, 'profile'  
]);
```

Available router methods/protocols:

```
//GET method
Route::get('/profile', [ ... ]);

//POST method
Route::post('/profile', [ ... ]);

//PUT method
Route::put('/profile', [ ... ]);

//PATCH method
Route::patch('/profile', [ ... ]);

//DELETE method
Route::delete('/profile', [ ... ]);
```


Extras

return redirect

```
Route::redirect('/here', '/there');
```

return permanent redirect

```
Route::permanentRedirect('/here', '/there');
```

return array/object

```
Route::get('/', function () {  
    return [1, 2, 3];  
});
```

return JSON

```
Route::get('/', function () {  
    return response()->json([  
        'name' => 'Ahmad Saiful Barhi',  
        'gender' => 'Male',  
        'email' => 'me@azbahri.my',  
    ]);  
});
```

Controller Location: **app/Http/Controllers/...**

```
$ php artisan make:controller <controller_name>
```

```
7  class HomeController extends Controller
8  {
9      /**
10       * Show the application dashboard.
11       *
12       * @return \Illuminate\Contracts\Support\Renderable
13       */
14     public function index()
15     {
16         return view('welcome');
17     }
18 }
```

```
$ php artisan make:controller <controller_name> --resource
```

Verb	URI	Action	Route Name
GET	/photos	index	photos.index
GET	/photos/create	create	photos.create
POST	/photos	store	photos.store
GET	/photos/{photo}	show	photos.show
GET	/photos/{photo}/edit	edit	photos.edit
PUT/PATCH	/photos/{photo}	update	photos.update
DELETE	/photos/{photo}	destroy	photos.destroy

Blade Location: **resources/views/...**

File name must end with **.blade.php**

```
@extends()
```

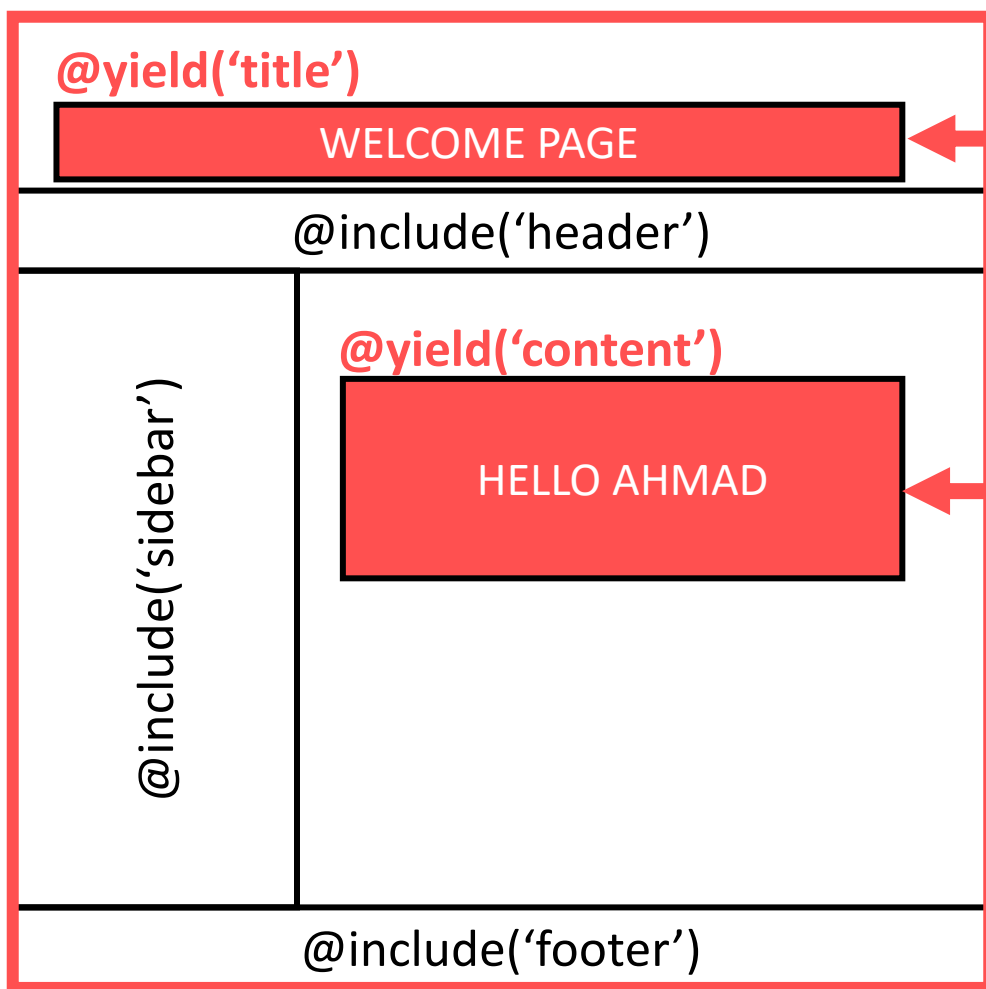
```
@section() ... @endsection
```

```
@include()
```

```
@if() ... @endif
```

```
@for() ... @endfor
```

```
{{ $variable }}
```



`layouts/admin.blade.php`

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

```
@section('title')
```

```
WELCOME PAGE
```

```
@endsection
```

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

```
HELLO AHMAD
```

```
@endsection
```

`pages/welcome.blade.php`

Grouping

```
Route::group([
    'prefix' => '/admin',
    'as' => 'admin.',
    'middleware' => []
], function () {

    //route 1
    //route 2
    ...

});
```

Dynamic URL

```
Route::get('/user/{user_id}', [
    App\Http\Controllers\UserController::class, 'view'
])->name('subscription.view');
```

Dynamic URL / with variable

```
Route::get('/user/{user_id}', [  
    App\Http\Controllers\UserController::class, 'view'  
])->name('subscription.view');
```

Receiving/accessing the variable

```
public function view($user_id)  
{  
    //variable $user_id is accessible here  
    //...  
}
```


Get URL information

```
// Get the current URL without the query string...  
url()->current();  
  
// Get the current URL including the query string...  
url()->full();  
  
// Get the full URL for the previous request...  
url()->previous();
```

Request

```
use Illuminate\Http\Request;

class UserController extends Controller {

    public function myFunction(Request $request)
    {
        $name = $request->input('name');
        //
    }

}
```

Retrieving All Input Data

```
//retrieve all request as an array
$input = $request->all();
```

```
//retrieve all request as a collection
$input = $request->collect();
```

HTML forms

```
<form method="POST" action="{ route('user.store') }">

    <input type="text" name="name">
    <input type="text" name="phone_no">
    <button type="submit">
        Submit
    </button>

</form>
```

CSRF

@csrf



```
<input type="hidden" name="_token" value="H0rGJ4MPOGCxGZVwOBFyTJcRniZCwhuESNIAc2hm">
```

Retrieving previous inputs

```
<input type="text" name="name" value="{ old('name') }">
```

Validation usage example

```
$this->validate($request, [  
    'type' => 'required',  
    'name' => 'required|min:3|max:60',  
    'email' => 'required|email|unique:users,email',  
    'password' => 'required|string|min:6',  
    'mobile' => 'nullable',  
]);
```

More validation rules available

email	- Check valid email address
integer	- 1, 2, 100, etc
numeric	- 1, 2, 0.55, etc
url	- Check valid URL
date	- Check valid date
date_format:YYYY-MM-DD	- Date with specific format
min:5	- At least 5 chars length
max:100	- Not more than 100 chars
required	- Field is compulsory / cannot be null
mimes:jpg,png,pdf	- Allow file type JPEG, PNG, or PDF only

Model Location: `app/Models/...`

Migration Location: `database/migrations/...`

Creating a new model

```
$ php artisan make:model <model_name>
```

Creating a new migration

```
$ php artisan make:migration <migration_name>
```

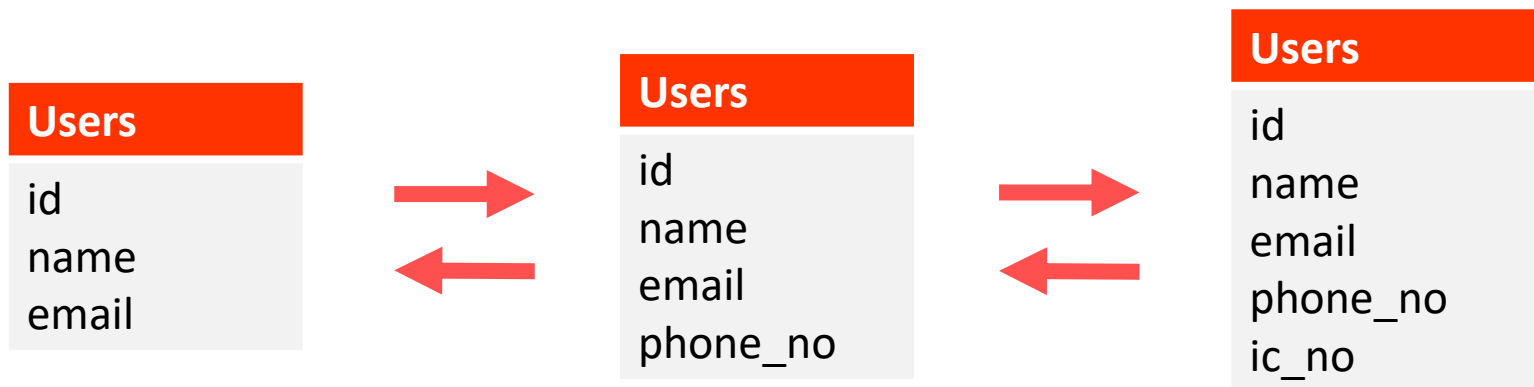
Creating a new model, together with a migration

```
$ php artisan make:model <model_name> -m
```

Migration Location: **database/migrations/...**

```
Schema::create('users', function (Blueprint $table) {  
    $table->id();  
    $table->string('name');  
    $table->string('email')->unique();  
    $table->timestamp('email_verified_at')->nullable();  
    $table->string('password');  
    $table->rememberToken();  
    $table->timestamps();  
});
```

Run the migration file `$ php artisan migrate`



Rollback the migration

```
$ php artisan migrate:rollback
```

Reset the migration

```
$ php artisan migrate:reset
```

Refresh the migration

```
$ php artisan migrate:refresh
```

Check the migration status

```
$ php artisan migrate:status
```


On Company Model: **app/Models/Company.php**

```
public function branches(){  
    return $this->hasMany('App\Models\Branch');  
}
```

On Branch Model: **app/Models/Branch.php**

```
public function company(){  
    return $this->belongsTo('App\Models\Company');  
}
```

Usage example

```
@foreach($company->branches as $branch)  
    {{ $branch->company->name }}  
@endforeach
```

Get the required package laravel/ui

```
$ composer require laravel/ui:^4.x
```

Choose either to use Vue or Bootstrap

```
$ php artisan ui vue --auth
```

```
$ php artisan ui bootstrap --auth
```

Once ready, run the command to generate UI scaffold

```
$ npm install && npm run dev
```

```
$ npm run build
```

 Or to build the asset file statically

Middleware Location: **app/Http/Middleware/...**

Create new middleware

```
$ php artisan make:middleware <middleware_name>
```

```
public function handle($request, Closure $next)
{
    $user = Auth::user();

    if($user && $user->age < 18){
        return redirect()->route('kids_section');
    } else {
        return $next($request);
    }
}
```

Registering middleware (**app/Http/Kernel.php**)

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

Applying middleware to route

```
Route::group(['middleware' => ['checkAge']], function() {  
    Route::get('/adult', 'Controller@index')->name('adult_section');  
});
```

Get a (single) record having `id` = 2

```
$article = Article::find(2);
```

Fetch all records

```
$articles = Article::all();
```

Fetch records with some basic query

```
$articles = Article::where('status', 'published')->orderBy('title', 'ASC')->get();
```

Count result

```
$total = Article::whereIn('status', ['published', 'draft'])->get()->count();
```

Collections with Pagination

```
$branches = Branch::where('company_id', $company->id)->paginate(10);
```

To display in blade view

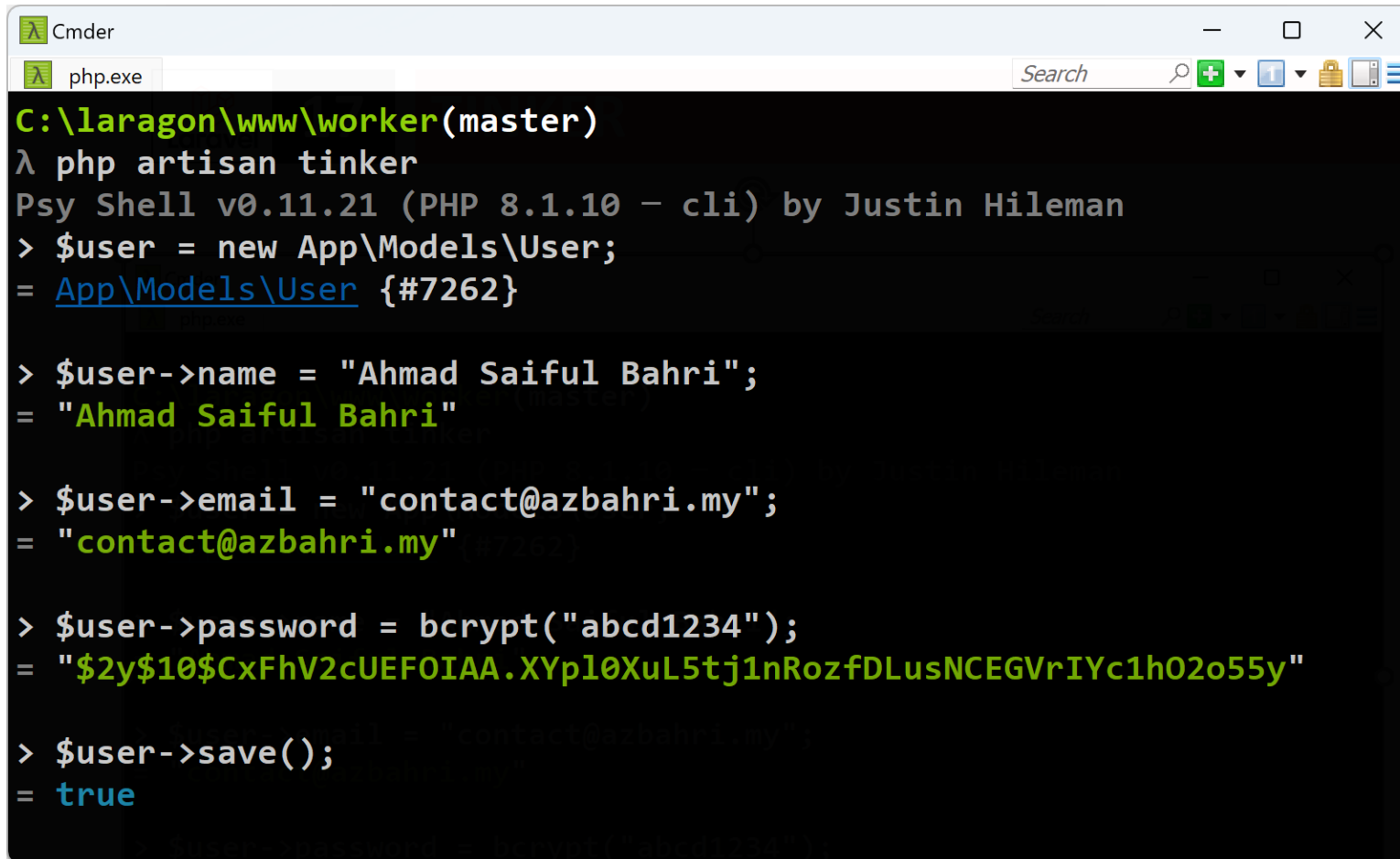
```
{!! $branches->appends($_GET)->render() !!}
```

Ask Laravel to use Bootstrap styling

```
public function boot()  
{  
    Paginator::useBootstrap();  
}
```

To start Tinker mode

```
$ php artisan tinker
```



```

C:\laragon\www\worker(master)
λ php artisan tinker
Psy Shell v0.11.21 (PHP 8.1.10 - cli) by Justin Hileman
> $user = new App\Models\User;
= App\Models\User {#7262}

> $user->name = "Ahmad Saiful Bahri";
= "Ahmad Saiful Bahri"

> $user->email = "contact@azbahri.my";
= "contact@azbahri.my"

> $user->password = bcrypt("abcd1234");
= "$2y$10$CxFhV2cUEFOIAA.XYp10XuL5tj1nRozfDLusNCEGVrIYc1h02o55y"

> $user->save();
= true
  
```

Install the package

```
$ composer require maatwebsite/excel
```

Create the export class

```
$ php artisan make:export ExportUser --model=User
```

```
namespace App\Exports;
use App\Models\User;
use Maatwebsite\Excel\Concerns\FromCollection;

class ExportUser implements FromCollection {
    public function collection()
    {
        return User::select('name','email')->get();
    }
}
```


Method/function in Controller (eg UserController)

```
public function exportUsers(Request $request){  
    return Excel::download(new ExportUser, 'users.xlsx');  
}
```

And lastly, don't forget to register the URL in Route

```
Route::get('/export', [UserController::class, 'exportUsers'])->name('export');
```



18

NOTES

A large, empty rectangular box with a thin gray border, intended for taking notes.



18

NOTES

A large, empty rectangular box with a thin gray border, intended for taking notes.



Q & A

Contact

Ahmad Saiful Bahri

me@azbahri.my

+60193327204

www.azbahri.my