

E-Commerce project initial setup commands

1. `composer create-project laravel/laravel ecommerce`

2. create database `logics\_builder\_ecommerce` default charset utf8 collate utf8_general_ci;
DB LEVEL USER AND PASSWORD
lb_ecommerce_user
ecommerce_pASS_4A

3. set .env file

4. `cd ecommerce`
`php artisan migrate`

5. install SPATIE LARAVEL PERMISSIONS
`composer require spatie/laravel-permission`

FORTIFY

6. `composer require laravel/fortify`

7. `php artisan vendor:publish --provider="Laravel\Fortify\FortifyServiceProvider"`

8. `composer require laravel/jetstream`

9. `php artisan jetstream:install livewire`

10. `npm install`
`npm run dev`

11. config/app.php file:
providers array
Spatie\Permission\PermissionServiceProvider::class,
You should publish the migration and the config/permission.php config file with:

12. `php artisan vendor:publish --provider="Spatie\Permission\PermissionServiceProvider"`

13. `php artisan migrate`

14. The `vendor:publish` command discussed above will also publish the `App\Providers\FortifyServiceProvider` class.
You should ensure this class is registered within the providers array of your application's config/app.php configuration file.

15. to clear cache and other config cache
`php artisan cache:clear`
`php artisan config:cache`
`php artisan route:clear`
`php artisan view:clear`

singleton design pattern (concept)

16. `vendor/laravel/fortify/routes`
file where fortify makes its own routes related to user login and logout

17. enclose all other routes in web.php inside
Route::middleware('auth:web', 'verified')->group(function () {

});

18. now register a user
name : your user name
email : admin@gmail.com
password : admin321
19. dashboard is in app/providers/RouteServiceProvider
public const HOME = '/dashboard';

20. where is the authentication
app/providers/FortifyServiceProvider
write the below two lines in the imports of the above class
use App\Models\User;
use Illuminate\Support\Facades\Hash;

in the boot method write the following code at the start

```
Fortify::authenticateUsing(function (Request $request) {  
    $user = User::where('email', $request->email)  
        ->first();  
    if ($user && (Hash::check($request->password, $user->password))) {  
        return $user;  
    }  
});
```

21. in app/providers/AuthServiceProviders
in the boot method after registerPolicies()
paste the following code

```
$this->registerPolicies();  
Gate::before(function ($user, $ability) {  
    return $user->hasRole('Super Admin') ? true : null;  
});
```

uncomment the use Illuminate\Support\Facades\Gate;
import line

22. in the app/http
make a new folder named Responses

23. paste LoginResponse.php and LogoutResponse.php file inside
search in <https://logics-builder.com/downloads>

24. in the import section of \vendor\laravel\fortify\src\Http\Controllers\AuthenticatedSessionController add
below lines
use App\Http\Responses\LoginResponse as CustomLoginResponse;
use App\Http\Responses\LogoutResponse as CustomLogoutResponse;

25. in vendor\laravel\fortify\src\Http\Controllers/AuthenticatedSessionController
inside store() method inside loginPipeline callback function replace the return line by following line
return app(CustomLoginResponse::class);

26. in the same AuthenticatedSessionController class destroy method write the following line
return app(CustomLogoutResponse::class);

27. now, lets register the following 2 roles
super admin, data entry

```
insert into `roles`(`name`, `guard_name`, `created_at`, `updated_at`)  
values  
( 'Super Admin', 'web', '2023-02-28 15:03:03', '2023-02-28 15:03:03'),  
( 'Data Entry', 'web', '2023-02-28 15:03:03', '2023-02-28 15:03:03');
```

28. now lets register the permissions Add Role, Block Unblock Role
insert into `permissions`(`name`, `guard\_name`, `created\_at`, `updated\_at`)
values
('Add Role', 'web', '2023-04-14 13:37:03', '2023-04-14 13:37:03'),
('Block Unblock Role', 'web', '2023-04-14 13:37:03', '2023-04-14 13:37:03');

29. role_has_permissions
2 is the id of data entry
insert into `role\_has\_permissions`(role_id, permission_id)
values(2, 1), (2, 2);

make migration for one super admin users

```
insert into `model_has_roles`(`role_id`, `model_type`, `model_id`)  
values(1, 'App\\Models\\User', 1);
```

30. now include
use HasRoles;
in user model class

also in imports section of User class add the below line
use Spatie\Permission\Traits\HasRoles;

31. to test permissions make a new controller with a method make a route
for it write below line in imports section of test controller class
use Illuminate\Support\Facades\Auth;

32. dbal package
composer require doctrine/dbal

33. laravel / auditing
composer require owen-it/laravel-auditing

34. Edit the config/app.php file and add the following line to register the service provider:

```
'providers' => [  
    // ...  
  
    OwenIt\Auditing\AuditingServiceProvider::class,  
  
    // ...  
],
```

35. Publishing

After configuring your framework of choice, use the following command to publish the configuration settings:

```
php artisan vendor:publish --provider "OwenIt\Auditing\AuditingServiceProvider" --tag="config"
```

This will create the config/audit.php configuration file.

36. Database

Publish the audits table migration to the database/ directory with the following command:

```
php artisan vendor:publish --provider "OwenIt\Auditing\AuditingServiceProvider" --tag="migrations"
```

37. php artisan migrate

38. FPDF Installation

Add to your "composer.json" file under the "require-dev" section:

```
"fpdf/fpdf": "^1.83"
```

then issue composer update command

39. Make a folder named Reports in app.

40. make a folder named Support inside Reports.

41. we will make LogicsBuilderPdf.php file inside Support

42. inside Reports folder, we will make IReport.php and AbstractReport.php files.

REPOSITORY PATTERN CONFIGURATION

43. make a folder named Repositories inside app folder.

44. we will need two files,

IBaseRepository.php

BaseRepository.php

to minimize code duplication for main CRUD operations.

45. where is the fortify login page.

it is inside resources\views\auth\login.blade.php file

46. add APP_TIMEZONE='Asia/Karachi' in .env file

47. Then, add the following in the app.php file:

```
'timezone' => env('APP_TIMEZONE', 'UTC'),
```

// The second parameter UTC is set as the default timezone value

48. php artisan config:clear

49. how to make validation rule in laravel (how to make custom validation rule)

muhammad ali (allowed)

muhammad ali 3310 (not allowed)

muhammad ali #123 (not allowed)

php artisan make:rule AlphaSpaceRule

it will be in app\Rules folder

DONE

next we will configure admin dashboard and front end website.

Basic setting has been completed.