



On line store for Handmade products

Technical
Documentation

PREPARED BY
LANA MUSTAFA
APR,19,2021

Main Goal::

Marketing handmade products online in an easy and effective way to support their suppliers and at the same time providing a wide and beautiful collection of handmade products for those interested.

The technologies that are covered by the project::

FRONT-END TECHNOLOGIES:

HTML , CSS ,
BOOTSTRAP AND
JAVA SCRIPT

BACK-END TECHNOLOGIES:

LARAVEL8 AND
MYSQL.

Project functional specifications(features)::

- Dashboard for Admin with two roles : (super admin and admin) to manage all resources.
- Customize dashboard for the supplier to manage his products.
- Public site consist of landing page , shop page , supplier profile , user profile , single page product , cart and checkout .
- Users can browse landing page , shop page , supplier profile and single page product without sign in or register .
- Supplier profile shows a description of him and collection of his products .
- To complete checkout user must be logged in .
- Login || Register system for both dashahboard and public site.

The most significant code excerpts and their justification::

```
use App\Http\Controllers\ProductController;
```

```
Route::resource('products',  
ProductController::class);
```

Using Laravel resource routing assigns the typical create, read, update, and delete ("CRUD") routes to a controller with a single line of code.

the test case of the most representative feature::

The user can browse the store and add any products to the cart without being logged in and when he wants to complete checkout by pressing the button he will be redirected to the login page to sign in first and then complete the checkout process and this applied by using middleware.

Description of the watch over security vulnerabilities::

Login with the same link for each user, supplier, and admin and this is less secure so it will be more secure by separating the login page for each of them and making different tables for them.

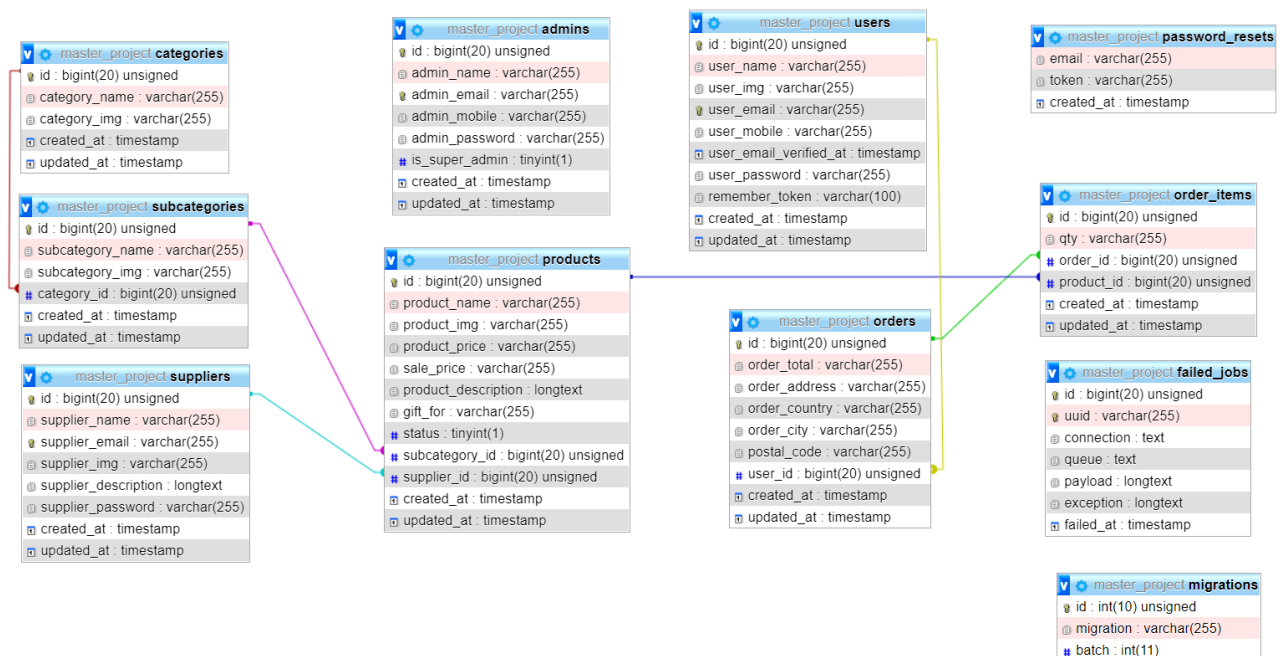
For security::

- Cross-Site Request Forgery (CSRF) Protection on Laravel.
- Hashing password .
- Authentication for user , supplier and admin (guard , session and middlewares).

Obstacles::

Using version8 which is a new and latest version in the laravel framework, although I learned version 5.8 during training so I had to check its documentation and google some of the things at the beginning to know how to do things and so on.

Database Diagram::



Laravel Structure::

```
Project Directory
|--app
    |--console
    |--Exceptions
    |--Http
    |--Providers
|--bootstrap
    |--Cache
|--config
|--database
|--public
    |--CSS
    |--fonts
    |--img
    |--js
|--console
|--Exceptions
|--Http
|--Providers
|--resources
    |--views
    |--layouts
|--partials
    |--footer-scripts.blade.php
    |--head.blade.php
    |--header.blade.php
    |--nav.blade.php
|--mainlayout.blade.php
|--routes
    |--api.php
    |--channels.php
    |--console.php
    |--web.php
|--storage
|--tests
|--vendor
|--editorconfig
|--env
|--env.example
|--styleci.yml
|--artisan
|--composer.json
|--composer.lock
|--package.json
|--package-lock.json
|--phpunit
|--README.md
|--webpack.mix
```