Criar projeto estoque com o comando

laravel new estoque

composer create-project laravel/laravel estoque\_laravel "5.1.\*"

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Alterar rota no routes\web.php para alterar rota do app web

Route::get('/', function () {

return "<h1>Listagem de produtos</h1>";

});

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Criar controller em app\Http\Controllers\ProdutoController.php

<?php namespace App\Http\Controllers;

class ProdutoController extends Controller{

public function lista(){

return '<h1>Listagem de produtos</h1>';

}

}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Agora criamos um rota para executar o método lista() do controller

Route::get('/produtos', 'ProdutoController@lista');

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Agora temos que listar o banco de dados em nosso laravel, vamos abrir o arquivo em config\database.php

Coloque o nome do banco nesse caso estoque, no trecho do arquivo e tbm o root no username

'mysql' => [

'driver' => 'mysql',

'host' => env('DB\_HOST', '127.0.0.1'),

'port' => env('DB\_PORT', '3306'),

'database' => env('DB\_DATABASE', 'estoque'),

'username' => env('DB\_USERNAME', ‘root’),

'password' => env('DB\_PASSWORD', ''),

'unix\_socket' => env('DB\_SOCKET', ''),

'charset' => 'utf8mb4',

'collation' => 'utf8mb4\_unicode\_ci',

'prefix' => '',

'strict' => true,

'engine' => null,

],

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Alteramos o método lista() do nosso ProdutoController.php para exibir as informações do banco na tela e adicionamos também o use DB

<?php namespace App\Http\Controllers;

use Illuminate\Support\Facades\DB;

class ProdutoController extends Controller

{

public function lista()

{

$produtos = DB::select('select \* from produtos');

dd($produtos);

return '<h1>Listagem de produtos</h1>';

}

}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Alteramos a configuração do nosso .env para

APP\_ENV=local

APP\_DEBUG=true

APP\_KEY=K8ERLYo8B0GGCaEFqv0ZfxBxvv6v70SG

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_DATABASE=estoque\_laravel

DB\_USERNAME=root

DB\_PASSWORD=

CACHE\_DRIVER=file

SESSION\_DRIVER=file

QUEUE\_DRIVER=sync

REDIS\_HOST=127.0.0.1

REDIS\_PASSWORD=null

REDIS\_PORT=6379

MAIL\_DRIVER=smtp

MAIL\_HOST=mailtrap.io

MAIL\_PORT=2525

MAIL\_USERNAME=null

MAIL\_PASSWORD=null

MAIL\_ENCRYPTION=null

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Agora vamos exbir os nossos registros de maneira mais organizada com o foreach

class ProdutoController extends Controller

{

public function lista()

{

$html = '<h1>Listagem de produtos</h1>';

$produtos = DB::select('select \* from produtos');

foreach($produtos as $p){

$html .= "<br/> Nome: " . $p->nome;

}

return $html;

}

}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Vamos instalar o scafold e ele já vai trazer as views para vc

composer require bestmomo/scafold

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Depois faça essa alteração no config\app.php

Bestmomo\Scafold\ScafoldServiceProvider::class,

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Agora vamos executar o publish para mostrar os arquivos que eu não tenho no meu projeto

php artisan vendor:publish

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Criar o arquivo listagem.php em resources\views

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Controle de estoque</title>

<link rel="stylesheet" href="">

</head>

<body>

<h1>Listagem de produtos</h1>

<table>

<?php foreach($produtos as $p): ?>

<tr>

<td><?= $p->nome ?></td>

<td><?= $p->valor ?></td>

<td><?= $p->descricao ?></td>

<td><?= $p->quantidade ?></td>

</tr>

<?php endforeach ?>

</table>

</body>

</html>

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Alterar o controller ProdutoController.php

class ProdutoController extends Controller

{

public function lista()

{

$produtos = DB::select('select \* from produtos');

return view('listagem')->with('produtos', $produtos);

}

}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Em seguida acesse o link para testar http://localhost:8000/produtos

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Altere o listagem.php

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Controle de estoque</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css" integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vslU6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M" crossorigin="anonymous">

<link rel="stylesheet" href="">

</head>

<body>

<h1>Listagem de produtos</h1>

<table class="table">

<?php foreach($produtos as $p): ?>

<tr>

<td><?= $p->nome ?></td>

<td><?= $p->valor ?></td>

<td><?= $p->descricao ?></td>

<td><?= $p->quantidade ?></td>

</tr>

<?php endforeach ?>

</table>

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js" integrity="sha384-b/U6ypiBEHpOf/4+1nzFpr53nxSS+GLCkfwBdFNTxtclqqenISfwAzpKaMNFNmj4" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/js/bootstrap.min.js" integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi047NrKGLO3SEJAg45jXxnGIfYzk4Si90RDIqNm1" crossorigin="anonymous"></script>

</body>

</html>

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Vamos instalar o bootstrap no Laravel adicione as pasta css e fonts com os arquivos em estoque\public\

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Agora criamos uma rota para a nossa lupa em listagem.php em routes.php

Route::get('/produtos/mostra','ProdutoController@mostra');

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Agora vamos criar o método mostra() em nosso controller ProdutoController.php

Use Request;

public function mostra()

{

$id = Request::route('id');

$produto = DB::select('select \* from produtos where id = ?', [$id]);

return view('detalhes')->with('p', $produto[0]);

}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Criamos o detalhes.php nas views

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Controle de estoque</title>

<link rel="stylesheet" type="text/css" href="/css/app.css">

<!-- <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css" integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vslU6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M" crossorigin="anonymous"> -->

</head>

<body>

<div class="container">

<h1>Detalhes do produtos <?= $p->nome ?></h1>

<ul>

<li>

Valor: <?= $p->valor ?>

</li>

<li>

Descrição: <?= $p->descricao ?>

</li>

<li>

Quantidade em estoque: <?= $p->quantidade ?>

</li>

</ul>

</div>

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js" integrity="sha384-b/U6ypiBEHpOf/4+1nzFpr53nxSS+GLCkfwBdFNTxtclqqenISfwAzpKaMNFNmj4" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/js/bootstrap.min.js" integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi047NrKGLO3SEJAg45jXxnGIfYzk4Si90RDIqNm1" crossorigin="anonymous"></script>

</body>

</html>

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Criamos uma rota mostra+id para acessar detalhes.php em routes.php

Route::get('/produtos/mostra/{id}','ProdutoController@mostra');

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Agora dentro da views vamos criar o principal.blade.php para deixa-lo com o conteúdo cabeçalho e rodapé podendo remover esse de outras estruturas adicionar o @yield onde ficará o conteúdo de outras paginas

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Controle de estoque</title>

<link rel="stylesheet" type="text/css" href="/css/app.css">

<!-- <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css" integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vslU6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M" crossorigin="anonymous"> -->

</head>

<body>

@yield('conteudo')

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js" integrity="sha384-b/U6ypiBEHpOf/4+1nzFpr53nxSS+GLCkfwBdFNTxtclqqenISfwAzpKaMNFNmj4" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/js/bootstrap.min.js" integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi047NrKGLO3SEJAg45jXxnGIfYzk4Si90RDIqNm1" crossorigin="anonymous"></script>

</body>

</html>

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Agora na renomear a pagina listagem.php para listagem.blade.php vamos adicionar a estrutura de template

@extends('principal')

@section('conteudo')

<div class="container">

<h1>Listagem de produtos</h1>

<table class="table">

<?php foreach($produtos as $p): ?>

<tr>

<td><?= $p->nome ?></td>

<td><?= $p->valor ?></td>

<td><?= $p->descricao ?></td>

<td><?= $p->quantidade ?></td>

<td>

<a href="/produtos/mostra/<?= $p->id ?>">

<span class="glyphicon glyphicon-search" aria-hidden="true">

</span>

</a>

</td>

</tr>

<?php endforeach ?>

</table>

</div>

@stop

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

A mesma coisa com a pagina detalher.php em detalhes.blade.php

@extends('principal')

@section('conteudo')

<div class="container">

<h1>Detalhes do produtos <?= $p->nome ?></h1>

<ul>

<li>

Valor: <?= $p->valor ?>

</li>

<li>

Descrição: <?= $p->descricao ?>

</li>

<li>

Quantidade em estoque: <?= $p->quantidade ?>

</li>

</ul>

</div>

@stop

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

O ProdutoController.php já alterado

public function mostra()

{

$id = Request::route('id');

$produto = DB::select('select \* from produtos where id = ?', [$id]);

return view('detalhes')->with('p', $produto[0]);

}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Com esse o laravel pode-se usar o {{ }} no lugar do <?= ?> deixando mais legível, pode ser feita de um jeito ou de outro

@extends('principal')

@section('conteudo')

<div class="container">

<h1>Detalhes do produtos {{ $p->nome }}</h1>

<ul>

<li>

Valor: R$ {{$p->valor}}

</li>

<li>

Descrição: {{$p->descricao}}

</li>

<li>

Quantidade em estoque: {{$p->quantidade}}

</li>

</ul>

</div>

@stop

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Caso o detalhes.blade.php tem um recurso para apresentar uma mensagem quando o campo estiver vazio como na opção abaixo

<li>

Descrição: {{$p->descricao or 'não têm descrição'}}

</li>

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Aplicando o recurso do laravel e deixando a sintaxe menos verbose

@extends('principal')

@section('conteudo')

<div class="container">

<h1>Listagem de produtos</h1>

<table class="table">

@foreach($produtos as $p)

<tr>

<td>{{ $p->nome }}</td>

<td>{{ $p->valor }}</td>

<td>{{ $p->descricao }}</td>

<td>{{ $p->quantidade }}</td>

<td>

<a href="/produtos/mostra/{{ $p->id }}">

<span class="glyphicon glyphicon-search" aria-hidden="true">

</span>

</a>

</td>

</tr>

@endforeach

</table>

</div>

@stop

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Adicionar condição no listagem.blade.php para adicionar class=”” ou class=”danger” caso menor que 1

<table class="table">

@foreach($produtos as $p)

<tr class="{{ $p->quantidade <= 1 ? 'danger' : ''}}">

<td>{{ $p->nome }}</td>

<td>{{ $p->valor }}</td>

<td>{{ $p->descricao }}</td>

<td>{{ $p->quantidade }}</td>

<td>

<a href="/produtos/mostra/{{ $p->id }}">

<span class="glyphicon glyphicon-search" aria-hidden="true">

</span>

</a>

</td>

</tr>

@endforeach

</table>

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Agora vamos implementar o insert começando pela rota em routes.php

Route::get('/produtos/novo','ProdutoController@novo');

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Em ProdutoController.php criamos a função novo()

public function novo()

{

return view('formulario');

}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Criamos o formulário.blade.php e adicionamos as estruturas padrão do blade

@extends('principal')

@section('conteudo')

@stop

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Adicione o conteúdo do formulário.blade.php

@extends('principal')

@section('conteudo')

<div class="container">

<form>

<div class="form-group">

<label>Nome</label>

<input class="form-control" />

</div>

<div class="form-group">

<label>Valor</label>

<input class="form-control"/>

</div>

<div class="form-group">

<label>Quantidade</label>

<input class="form-control"/>

</div>

<div class="form-group">

<label>Descrição</label>

<input class="form-control"/>

</div>

<button class="btn btn-primary" type="submit">Adicionar</button>

</form>

</div>

@stop

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Vamos criar uma rota para adicionar o produto em routes.php

Route::get('/produtos/adiciona','ProdutoController@adiciona');

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Vamos criar o método adicionar() em ProdutoController

public function adiciona()

{

return view('adicionado')->with('nome', 'Frigobar');

}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Criar arquivo na view o adicionado.blade.php

@extends('principal')

@section('conteudo')

Produto {{$nome}} adicionado com sucesso!

@stop

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Alteramos nosso formulário.blade.php

@extends('principal')

@section('conteudo')

<div class="container">

<form action="/produtos/adiciona">

<div class="form-group">

<label>Nome</label>

<input name="nome" class="form-control" />

</div>

<div class="form-group">

<label>Valor</label>

<input name="valor" class="form-control"/>

</div>

<div class="form-group">

<label>Quantidade</label>

<input name="quantidade" class="form-control"/>

</div>

<div class="form-group">

<label>Descrição</label>

<input name="descricao" class="form-control"/>

</div>

<button class="btn btn-primary" type="submit">Adicionar</button>

</form>

</div>

@stop

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Preparamos nosso método adiciona() para adicionar os campos do formulário no banco

public function adiciona()

{

//pegar as informações do formulario

$nome = Request::input('nome');

$valor = Request::input('valor');

$quantidade = Request::input('quantidade');

$descricao = Request::input('descricao');

//salvar no banco de dados

DB::insert('insert into produtos (nome, quantidade, valor, descricao) values(?, ?, ?, ?)', array($nome, $quantidade, $valor, $descricao));

return view('adicionado')->with('nome', $nome);

}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Alteramos o form adicionando o method=”post” para que esse form seja enviado como post, adicionando tbm o TOKEN para evitar ataque CSRF

@extends('principal')

@section('conteudo')

<div class="container">

<form action="/produtos/adiciona" method="post">

//precise ser o name \_token para identificar

<input type="hidden" name="\_token" value="{{ csrf\_token() }}" />

<div class="form-group">

<label>Nome</label>

<input name="nome" class="form-control" />

</div>

<div class="form-group">

<label>Valor</label>

<input name="valor" class="form-control"/>

</div>

<div class="form-group">

<label>Quantidade</label>

<input name="quantidade" class="form-control"/>

</div>

<div class="form-group">

<label>Descrição</label>

<input name="descricao" class="form-control"/>

</div>

<button class="btn btn-primary" type="submit">Adicionar</button>

</form>

</div>

@stop

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Alterar a rota para post deixando

Route::post('/produtos/adiciona','ProdutoController@adiciona');

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Vamos alterar para assim que inserido o novo registro ele redirecione para a pagina de listagem.blade.php , vamos alterar primeiro em nosso Produtocontroller.php no método adiciona() , usando tbm o ->withInput para dados da requisição anterior e adiciona a msg do if

public function adiciona()

{

//pegar as informações do formulario

$nome = Request::input('nome');

$valor = Request::input('valor');

$quantidade = Request::input('quantidade');

$descricao = Request::input('descricao');

//salvar no banco de dados

DB::insert('insert into produtos (nome, quantidade, valor, descricao) values(?, ?, ?, ?)', array($nome, $quantidade, $valor, $descricao));

//com a função withInput() mantem os parametros da função anterior;

return redirect('/produtos')->withInput();

}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Vamos adicionar o old para pegar o da requisição a coloca-la no nosso if do listagem.blade.php

@extends('principal')

@section('conteudo')

<div class="container">

<h1>Listagem de produtos</h1>

<table class="table">

@foreach($produtos as $p)

<tr class="{{ $p->quantidade <= 1 ? 'danger' : ' '}}">

<td>{{ $p->nome }}</td>

<td>{{ $p->valor }}</td>

<td>{{ $p->descricao }}</td>

<td>{{ $p->quantidade }}</td>

<td>

<a href="/produtos/mostra/{{ $p->id }}">

<span class="glyphicon glyphicon-search" aria-hidden="true">

</span>

</a>

</td>

</tr>

@endforeach

</table>

@if(old('nome'))

<div class="alert alert-success">

<!-- mantém os dados da requisição anterior -->

Produto {{old('nome')}} adicionado com sucesso!

</div>

@endif

</div>

@stop

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++